

(2)

NAVAL POSTGRADUATE SCHOOL

Monterey , California

AD-A257 329



DTIC
ELECTED
NOV 23 1992
S A D THEESIS

CORRECTION OF INERTIAL MEASUREMENTS

USING GPS UPDATES FOR UNDERWATER NAVIGATION

by

Steve Nagengast

September, 1992

Thesis Advisor:

Roberto Cristi

Approved for public release; distribution is unlimited.

92 11 20 025

92-29925



UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE

REPORT DOCUMENTATION PAGE				Form Approved OMB No 0704-0188
1a REPORT SECURITY CLASSIFICATION UNCLASSIFIED		1b RESTRICTIVE MARKINGS		
2a SECURITY CLASSIFICATION AUTHORITY		3 DISTRIBUTION/AVAILABILITY OF REPORT Approved for public release; distribution is unlimited.		
2b DECLASSIFICATION/DOWNGRADING SCHEDULE				
4 PERFORMING ORGANIZATION REPORT NUMBER(S)		5 MONITORING ORGANIZATION REPORT NUMBER(S)		
6a NAME OF PERFORMING ORGANIZATION Naval Postgraduate School	6b OFFICE SYMBOL (If applicable) 32	7a NAME OF MONITORING ORGANIZATION Naval Postgraduate School		
6c ADDRESS (City, State, and ZIP Code) Monterey, CA 93943-5000		7b ADDRESS (City, State, and ZIP Code) Monterey, CA 93943-5000		
8a NAME OF FUNDING/SPONSORING ORGANIZATION Naval Air Warfare Center	8b OFFICE SYMBOL (If applicable) 3333	9 PROCUREMENT INSTRUMENT IDENTIFICATION NUMBER		
8c ADDRESS (City, State, and ZIP Code) Pt. Mugu, CA 93042-5000		10 SOURCE OF FUNDING NUMBERS		
		PROGRAM ELEMENT NO	PROJECT NO	TASK NO
		WORK UNIT ACCESSION NO		
11 TITLE (Include Security Classification) CORRECTION OF INERTIAL MEASUREMENTS USING GPS UPDATES FOR UNDERWATER NAVIGATION				
12 PERSONAL AUTHOR(S) Nagengast, Steven John; Naval Air Warfare Center, Weapons Division				
13a TYPE OF REPORT Master's Thesis	13b TIME COVERED FROM _____ TO _____	14 DATE OF REPORT (Year, Month, Day) 1992, September	15 PAGE COUNT 104	
16 SUPPLEMENTARY NOTATION The views expressed in this thesis are those of the author and do not reflect the official policy or position of the Department of Defense or the U.S. Government.				
17 COSATI CODES		18 SUBJECT TERMS (Continue on reverse if necessary and identify by block number) Navigation, Global Positioning System, Kalman Filter, Underwater Vehicles, Inertial Navigation		
19 ABSTRACT (Continue on reverse if necessary and identify by block number) An Autonomous Underwater Vehicle (AUV) can combine a Global Positioning System (GPS) receiver with an Inertial Navigation System (INS) to navigate with a specified accuracy level. The AUV would be required to surface periodically to obtain a GPS fix. A computer simulation has been developed using an AUV model and an INS error model to generate noisy measurements. A Kalman filter is used to estimate the simulated INS errors. Several runs were executed to compare combinations of equipment with different levels of accuracy.				
20 DISTRIBUTION/AVAILABILITY OF ABSTRACT <input checked="" type="checkbox"/> UNCLASSIFIED/UNLIMITED <input type="checkbox"/> SAME AS RPT <input type="checkbox"/> DTIC USERS		21 ABSTRACT SECURITY CLASSIFICATION UNCLASSIFIED		
22a NAME OF RESPONSIBLE INDIVIDUAL Dr. Roberto Cristi		22b TELEPHONE (Include Area Code) (408) 646-2223	22c OFFICE SYMBOL EC/Cx	

Approved for public release; distribution is unlimited.

Correction of Inertial Measurements Using GPS
Updates for Underwater Navigation

by

Steve Nagengast
Naval Air Warfare Center, Weapons Division
B. S., California State Polytechnic University, 1985

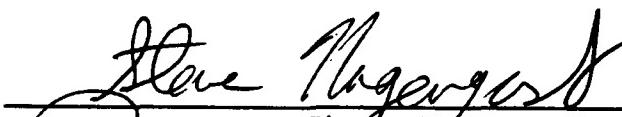
Submitted in partial fulfillment
of the requirements for the degree of

MASTER OF SCIENCE IN ELECTRICAL ENGINEERING

from the

NAVAL POSTGRADUATE SCHOOL
September, 1992

Author:



Steve Nagengast

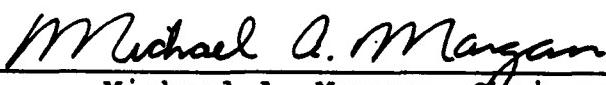
Approved by:



Roberto Cristi, Thesis Advisor



Harold A. Titus, Second Reader



Michael A. Morgan, Chairman
Department of Electrical and Computer Engineering

ABSTRACT

An Autonomous Underwater Vehicle (AUV) can combine a Global Positioning System (GPS) receiver with an Inertial Navigation System (INS) to navigate with a specified accuracy level. The AUV would be required to surface periodically to obtain a GPS fix. A computer simulation has been developed using an AUV model and an INS error model to generate noisy measurements. A Kalman filter is used to estimate the simulated INS errors. Several runs were executed to compare combinations of equipment with different levels of accuracy.

DTIC QUALITY INSPECTED 4

Accession For	
NTIS	CRA&I <input checked="" type="checkbox"/>
DTIC	TAB <input type="checkbox"/>
Unannounced <input type="checkbox"/>	
Justification	
By	
Distribution /	
Availability Codes	
Dist	Avail and/or Special
A-1	

TABLE OF CONTENTS

I.	INTRODUCTION	1
II.	GLOBAL POSITIONING SYSTEM	8
	A. LEVELS OF ACCURACY	8
	B. POSITION DETERMINATION	10
	C. REPRESENTATIVE GPS DATA	13
III.	INERTIAL NAVIGATION SYSTEMS	16
	A. GIMBALED SYSTEMS	16
	B. STRAPDOWN SYSTEMS	18
	C. GYROSCOPE PERFORMANCE CHARACTERISTICS	24
	D. OPTICAL GYROSCOPES	26
	1. Advantages of Optical Gyroscopes	26
	2. Theory of Operation: Sagnac Effect	28
	3. Ring Laser Gyros	31
IV.	INS ERROR ESTIMATION	35
	A. INS ERROR MODEL	35
	B. KALMAN FILTER EQUATIONS	37
V.	COMPUTER SIMULATION	41
VI.	RESULTS	48
VII.	CONCLUSIONS AND RECOMMENDATIONS	70
APPENDIX A.	GPS DATA PROCESSING SOFTWARE	72
APPENDIX B.	COMPUTER SIMULATION SOFTWARE	74
LIST OF REFERENCES		88
BIBLIOGRAPHY		90
INITIAL DISTRIBUTION LIST		91

LIST OF TABLES

Table 2.1:	ACCURACY LEVELS OF THE STANDARD POSITIONING SERVICE (SPS) AND THE PRECISE POSITIONING SERVICE (PPS)	9
Table 3.1:	FAMILY OF GYROSCOPE APPLICATIONS	26
Table 4.1:	INITIAL VARIANCES IN THE ERROR COVARIANCE MATRIX	40
Table 5.1:	STATE VARIABLE VARIANCES	45
Table 5.2:	ACCURACY LEVELS DURING THE COMPUTER SIMULATIONS.	47

LIST OF FIGURES

Figure 1.1:	Open-loop aiding of Inertial Navigation System (INS) measurements using Global Positioning System (GPS) data	4
Figure 1.2:	Closed-loop aiding of INS measurements using GPS data.	5
Figure 1.3:	Integrated GPS/INS.	6
Figure 2.1:	Horizontal errors in GPS position measurements.	14
Figure 2.2:	Vertical errors in GPS position measurements.	15
Figure 2.3:	Errors in GPS velocity measurements	15
Figure 3.1:	Reference frames used for inertial navigation systems.	17
Figure 3.2:	Local-level frame mechanization of INS measurement processing.	19
Figure 3.3:	Earth-fixed frame mechanization of an INS .	19
Figure 3.4:	Errors in an INS with a gyro drift of 0.01 deg/hr and an accelerometer bias of 0.0001 m/s ²	24
Figure 3.5:	Illustration of the Sagnac effect	29
Figure 5.1:	Block diagram of computer simulation.	41
Figure 5.2:	Block diagram of INS error model.	43
Figure 6.1:	AUV depth during initial surface interval .	48
Figure 6.2:	AUV depth during first five dives	48
Figure 6.3:	First-run, surface-interval attitude error.	49
Figure 6.4:	First-run, surface-interval position error.	49
Figure 6.5:	First-run, surface-interval velocity error.	50
Figure 6.6:	First-run, surface-interval gyro drift. . .	50
Figure 6.7:	First-run, surface-interval accelerometer bias.	50

Figure 6.8:	First-run, dive-series attitude error.	51
Figure 6.9:	First-run, dive-series position error.	51
Figure 6.10:	First-run, dive-series velocity error.	51
Figure 6.11:	First-run, dive-series gyro drift.	52
Figure 6.12:	First-run, dive-series accelerometer bias.	52
Figure 6.13:	First-run attitude error estimate error.	52
Figure 6.14:	First-run position error estimate error.	53
Figure 6.15:	First-run velocity error estimate error.	53
Figure 6.16:	First-run gyro drift estimate error.	53
Figure 6.17:	First-run accelerometer bias estimate error.	54
Figure 6.18:	Second-run, surface-interval attitude error.	54
Figure 6.19:	Second-run, surface-interval position error.	54
Figure 6.20:	Second-run, surface-interval velocity error.	55
Figure 6.21:	Second-run, surface-interval gyro drift.	55
Figure 6.22:	Second-run, surface-interval accelerometer bias	55
Figure 6.23:	Second-run, dive-series attitude error	56
Figure 6.24:	Second-run, dive-series position error	56
Figure 6.25:	Second-run, dive-series velocity error	56
Figure 6.26:	Second-run, dive-series gyro drift	57
Figure 6.27:	Second-run, dive-series accelerometer bias	57
Figure 6.28:	Second-run attitude error estimate error	57
Figure 6.29:	Second-run position error estimate error	58
Figure 6.30:	Second-run velocity error estimate error	58
Figure 6.31:	Second-run gyro drift estimate error	58

Figure 6.32:	Second-run accelerometer bias estimate error.	59
Figure 6.33:	Third-run, surface-interval attitude error	59
Figure 6.34:	Third-run, surface-interval position error	59
Figure 6.35:	Third-run, surface-interval velocity error	60
Figure 6.36:	Third-run, surface-interval gyro drift . .	60
Figure 6.37:	Third-run, surface-interval accelerometer bias	60
Figure 6.38:	Third-run, dive-series attitude error. . .	61
Figure 6.39:	Third-run, dive-series position error. . .	61
Figure 6.40:	Third-run, dive-series velocity error. . .	61
Figure 6.41:	Third-run, dive-series gyro drift.	62
Figure 6.42:	Third-run, dive-series accelerometer bias.	62
Figure 6.43:	Third-run attitude error estimate error. .	62
Figure 6.44:	Third-run position error estimate error. .	63
Figure 6.45:	Third-run velocity error estimate error. .	63
Figure 6.46:	Third-run gyro drift estimate error.	63
Figure 6.47:	Third-run accelerometer bias estimate error.	64
Figure 6.48:	Fourth-run, surface-interval attitude error.	64
Figure 6.49:	Fourth-run, surface-interval position error.	64
Figure 6.50:	Fourth-run, surface-interval velocity error.	65
Figure 6.51:	Fourth-run, surface-interval gyro drift. .	65
Figure 6.52:	Fourth-run, surface-interval accelerometer bias	65
Figure 6.53:	Fourth-run, dive-series attitude error . .	66
Figure 6.54:	Fourth-run, dive-series position error . .	66

Figure 6.55:	Fourth-run, dive-series velocity error . . .	66
Figure 6.56:	Fourth-run, dive-series gyro drift	67
Figure 6.57:	Fourth-run, dive-series accelerometer bias	67
Figure 6.58:	Fourth-run attitude error estimate error .	67
Figure 6.59:	Fourth-run position error estimate error .	68
Figure 6.60:	Fourth-run velocity error estimate error .	68
Figure 6.61:	Fourth-run gyro drift estimate error . . .	68
Figure 6.62:	Fourth-run accelerometer bias estimate error.	69

TABLE OF SYMBOLS

P, \mathbf{x} ~ position
 V, \mathbf{v} ~ velocity
 δp ~ INS position error
 δv ~ INS velocity error
 ϵ ~ INS attitude error
 ϵ_p ~ GPS position error
 ϵ_v ~ GPS velocity error
 $\hat{\epsilon}$ ~ estimate of ϵ
 $\tilde{\epsilon}$ ~ estimate error
 σ ~ standard deviation
 p ~ pseudorange measurement
 Ψ ~ noiseless pseudorange
 v_p ~ measurement noise
 β_p ~ time-correlated errors
 $[X_i, Y_i, Z_i]^T$ ~ position of satellite i
 $\mathbf{x} = [x, y, z]^T$ ~ user's position
 Δt ~ receiver's clock offset
 c ~ speed of light
 $\mathbf{x}_o = [x_o, y_o, z_o]^T$ ~ approximate user's position
 b ~ body frame
 e ~ earth-fixed frame
 n ~ local-level frame
 i ~ inertial frame
 ϕ ~ latitude
 λ ~ longitude
 h ~ altitude
 γ ~ normal gravity
 \mathbf{f}^b ~ specific force in b-frame coordinates
 R^n_b ~ transformation matrix from b- to n-frame
 \dot{R} ~ change in the R matrix
 $\dot{\mathbf{Y}}$ ~ acceleration
 ω_{ib}^b ~ angular rotation rate of the b-frame with respect to
 the i-frame, given in b-frame coordinates
 Ω_{ib}^b ~ skew-symmetric matrix of ω_{ib}^b
 $[i, j, k]^T$ ~ unit vector
 ω_e ~ earth's rotation rate
 $(\omega_{ob}^b)_1$ ~ roll rate
 $(\omega_{ob}^b)_2$ ~ pitch rate
 $(\omega_{ob}^b)_3$ ~ yaw rate
 k ~ sample number
 \mathbf{g}^e ~ gravity vector

$p(t)$ ~ position as a function of time
 $v(t)$ ~ velocity as a function of time
 b , a ~ accelerometer bias
 d , g ~ gyro drift
 R ~ radius of FOG coil
 N ~ number of turns in FOG coil
 A ~ area of FOG coil
 R_e ~ earth's radius
 ω_s ~ Schuler frequency
 τ ~ transit time
 Ω ~ FOG rotation rate
 ΔL ~ optical path length difference
 v ~ optical frequency
 L ~ optical path length
 λ ~ optical wavelength
 N ~ net number of accumulated fringe counts in a Sagnac interferometer
 θ ~ net angle turned
 I ~ fringe pattern intensity
 I_o ~ average intensity
 $\Delta\omega$ ~ angular beat frequency
 λ/ϵ ~ fringe spacing
 x ~ distance along fringe pattern
 ϕ ~ arbitrary phase angle
 \mathbf{x} ~ state vector
 w ~ white noise
 $[F]$ ~ dynamics matrix
 $[I]$ ~ identity matrix
 $[0]$ ~ null matrix
 F^* ~ skew-symmetric matrix of f^*
 ζ ~ Gauss-Markov constant for drift
 β ~ Gauss-Markov constant for bias
 G ~ system noise matrix
 $\Gamma(K)$ ~ discrete version of G
 $\Phi(k)$ ~ transition matrix
 $\mathbf{z}(k)$ ~ measurements
 t_k ~ sample time
 $H(k)$ ~ observation matrix
 $y(k)$ ~ measurement noise
 $Q(k)$ ~ system noise covariance matrix
 $R(k)$ ~ measurement noise covariance matrix
 $K(k)$ ~ Kalman gain matrix
 $P(k)$ ~ error covariance matrix
 $P^-(k)$ ~ initial estimate of $P(k)$

I. INTRODUCTION

An Autonomous Underwater Vehicle (AUV) is a submersible designed to operate independently of real-time human control. In order to carry on planned missions, such a vehicle requires a navigation system to determine its location at any time. For covert and long range operations, passive navigation is preferable, and has to be designed to provide sufficient accuracy during the entire mission.

A passive and accurate navigation system can be developed by combining an Inertial Navigation System (INS) with a Global Positioning System (GPS) receiver in a complementary fashion. Since the errors in any INS grow with time, they can be corrected by programming the AUV to surface periodically to get a GPS fix.

The GPS uses satellites to determine position and velocity. A measurement of the time-of-arrival of the GPS signal at the receiver is combined with knowledge of the satellite's position to estimate the range to the satellite and the user's clock error. This measurement is referred to as the "pseudorange". Another measurement, called the "delta range", is used to determine the user's velocity from the Doppler shift of the GPS carrier frequency. The accuracy of velocity determined by GPS is based on knowledge

of the orbital parameters of the satellites and the short carrier wavelength, which is approximately 19 cm [Ref. 1].

The accuracy of GPS position data depends on several factors, which are discussed in Chapter 2. There are differences, in terms of accuracy, between military and civilian applications, authorized and non-authorized users, respectively. An independent authorized user can determine position with a two-dimensional (2-D), horizontal accuracy of better than 17.8 meters [Ref. 2]. Non-authorized users can obtain commercially available equipment that provides independent position estimates with a 2-D accuracy on the order of 100 meters. If a reference station is available, then differential corrections can be used to provide accuracy of better than five meters for any user.

An INS uses accelerometers to measure the forces acting on a vehicle, and gyroscopes ("gyros") to determine the direction of those forces. Vehicle accelerations can be derived from the measured forces. By integrating the accelerations we can compute velocities, and with a second integration we obtain the position relative to the initial conditions. An INS also outputs the orientation of the vehicle, because this attitude information is required when determining the force directions.

The accuracy of INS data depends on the quality of the equipment. The trade-offs for better accuracy typically involve size, weight, and cost.

An INS can be used to measure a vehicle's trajectory. One problem with using an INS is the growth of errors over time due to the integration of acceleration. This drawback can be overcome by combining the INS with independent position and velocity measurements, in order to estimate and correct the INS errors.

This thesis addresses the use of a combination of INS and GPS equipment for the navigation of an AUV. Since GPS signals cannot be received underwater, GPS position and velocity measurements are obtained by requiring the vehicle to surface periodically. The integration of INS and GPS data is implemented using Kalman filtering techniques.

The Kalman filter approach described in this research is based on statistical models of the errors inherent to INS and GPS measurements. In particular, the INS gives measurements of position, velocity, and angular orientation of the vehicle in different coordinate frames (inertial and body fixed), while the GPS yields measurements of position and velocity in an earth-fixed frame. The two systems are combined as shown in Figure 1.1, where we can see that the Kalman filter attempts to determine an optimal estimate of the errors. The estimates are consequently used to correct one of the measurement sets (the INS, for example). This configuration is an open-loop implementation. The task of determining error models is of fundamental importance in this problem.

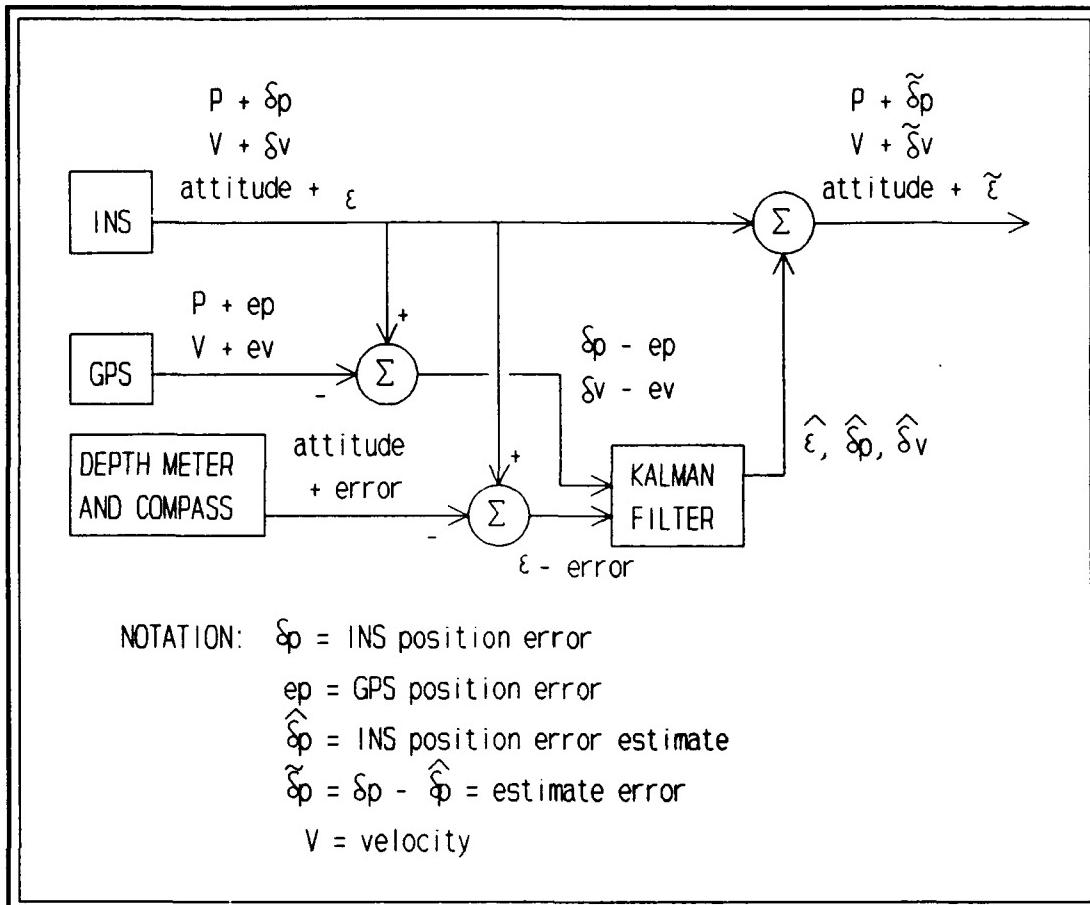


Figure 1.1: Open-loop aiding of Inertial Navigation System (INS) measurements using Global Positioning System (GPS) data. (After Ref. 3:p. 266)

The following approach can be used to determine the attitude errors. To obtain the error in pitch angle, a sample is taken when the depth rate sensor indicates zero vertical velocity. Assuming the AUV is in transition from a climb to a dive at this instant, the pitch angle is zero and the angle indicated by the INS is the error. The roll angle error is obtained by averaging samples of the roll angle indicated while the vehicle is near the surface. This error measurement is based on the assumption that the vehicle is

inherently stable and the roll angle will average zero during this interval. The heading error can be obtained by comparing a compass reading to the INS indication.

Figure 1.2 illustrates the alternative closed-loop implementation, which feeds the error estimates back into the INS. The INS then uses the error estimates to compute corrected position, velocity, and attitude measurements. This leads to a filter that linearizes about an estimated trajectory that is updated with each aiding measurement. This approach is called extended Kalman filtering and is the preferred method for long duration missions on the order of weeks or more [Ref. 3:pp. 356-379].

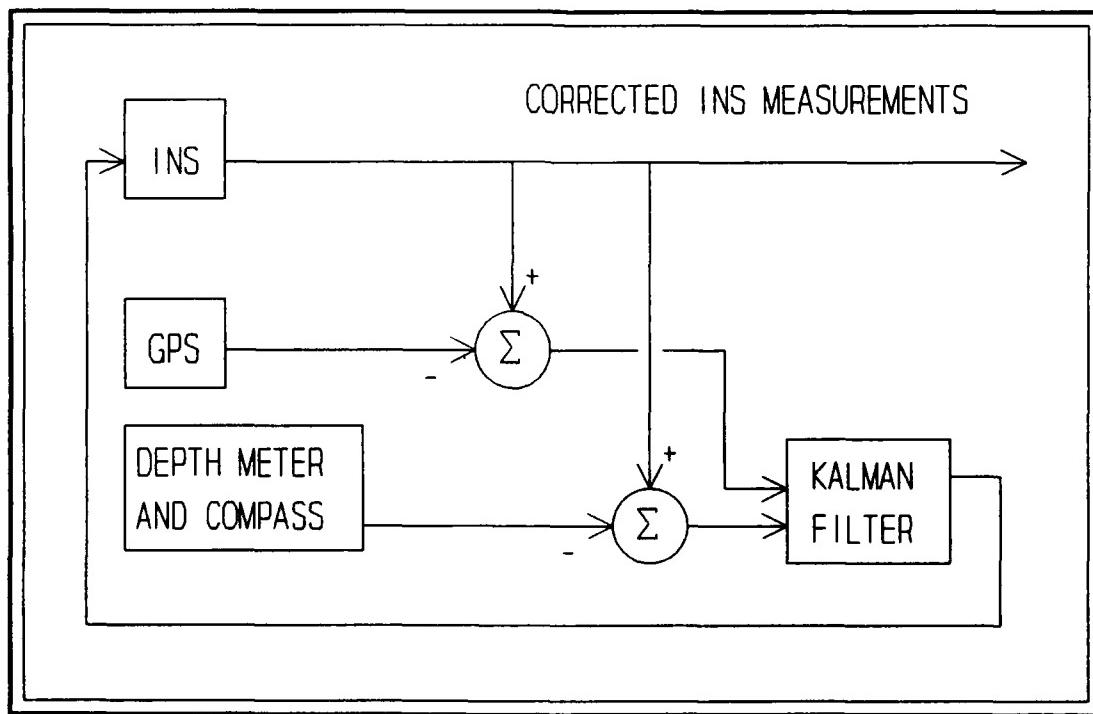


Figure 1.2: Closed-loop aiding of INS measurements using GPS data. (After Ref. 3:p. 369)

Both versions of GPS-aiding of an INS use position and velocity estimates from a Kalman filter within the GPS unit. The integration of GPS and INS measurements can be accomplished with a single Kalman filter, as illustrated in Figure 1.3 [Ref. 4].

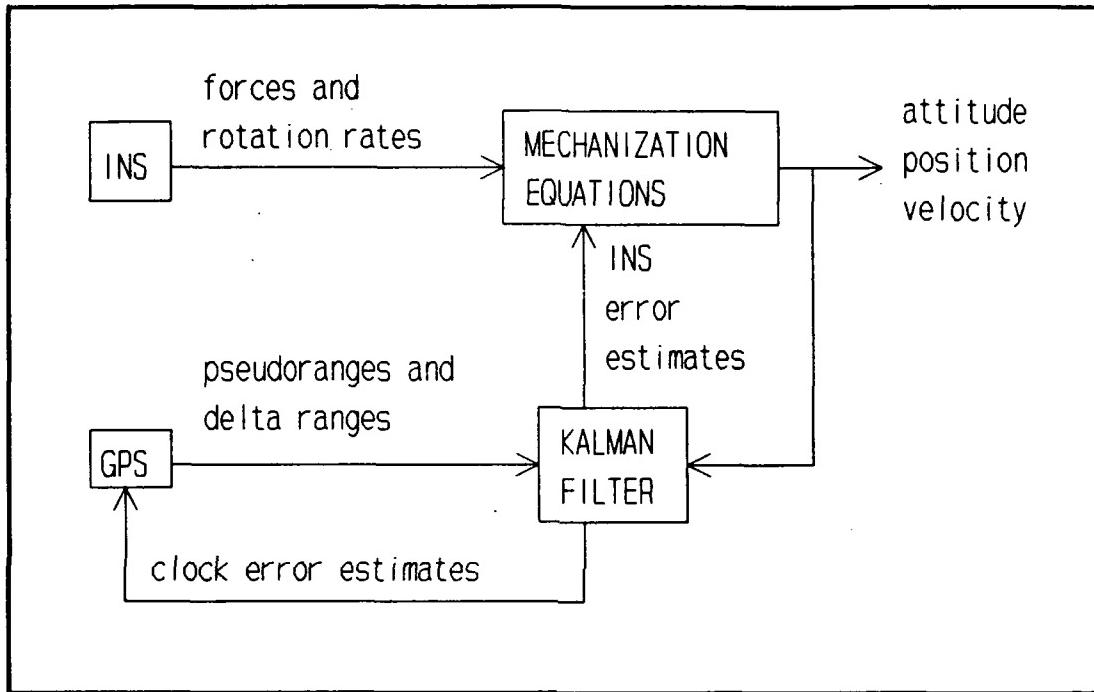


Figure 1.3: Integrated GPS/INS. (After Ref. 4)

This thesis focuses on using the open-loop approach to aiding an INS with GPS. In this case the Kalman filter linearizes about the reference trajectory provided by the INS. Chapter II discusses GPS in terms of the available levels of accuracy and the procedure for determining position from the raw measurements. Some representative GPS data is included in chapter II to illustrate the typical errors inherent to the system.

In chapter III, the various types of INS's are described and background information is provided to clarify the terminology. The INS error model is developed in chapter IV, along with the Kalman filter equations.

The computer simulation is described in chapter V, and the results of several runs are presented in chapter VI. Finally, chapter VII provides conclusions and recommendations for further work in this area. Software listings are included in the appendices, along with a bibliography, to assist with related research. The software has not been verified beyond the results documented here and further application is the responsibility of the user.

II. GLOBAL POSITIONING SYSTEM

A. LEVELS OF ACCURACY

The GPS is a Department of Defense (DOD) satellite navigation system. The primary purpose of the GPS is to support military navigation requirements. Two levels of accuracy are being provided by the GPS in accordance with DOD policy. Authorized users, mostly military, are allowed to use the high-accuracy capabilities of the system. Some applications outside the military have been granted permission to use these capabilities. All other users are referred to as non-authorized users, but they have unlimited access to the low-accuracy capabilities.

Authorized users are provided with a high-accuracy navigation system that is totally passive, i.e., it does not require radiating signals from the user. The advantages of non-radiating user equipment include less power required, smaller size units, and the ability to operate covertly. These advantages are all due to the absence of a transmitter.

The use of the GPS by non-authorized users is being permitted at reduced accuracy levels. The techniques used for degrading the accuracy are referred to as Selective Availability (SA). The actual accuracy levels provided can be adjusted by the DOD in accordance with policy. In fact,

SA can be turned off entirely. The current policy (April 1992), states that two times the standard deviation of horizontal position will be less than or equal to 100 meters [Ref. 2]. The Greek letter "sigma" is commonly used to denote the standard deviation, so this measure of accuracy is referred to as the "two sigma" (2σ) level. The horizontal position is the standard 2-D solution, because it is the usual type of position required by ships at sea.

The authorized users will have access to the Precise Positioning Service (PPS), whereas non-authorized users will be limited to the Standard Positioning Service (SPS). The accuracy levels associated with SPS and PPS are compared in Table 2.1.

Table 2.1: ACCURACY LEVELS OF THE STANDARD POSITIONING SERVICE (SPS) AND THE PRECISE POSITIONING SERVICE (PPS)
(After Ref. 2)

	SPS	PPS
HORIZONTAL POSITION (2σ)	100 m	17.8 m
VERTICAL POSITION (2σ)	156 m	27.7 m
RECEIVER CLOCK SYNCHRONIZATION (1σ)	167 ns	100 ns

Another difference between SPS and PPS, in addition to SA, is the signal structure. For authorized users, a Precision code (P-code) is broadcast on two frequencies, L_1 and L_2 . A Coarse/Acquisition code (C/A code) on L_1 provides a coarse navigation capability to SPS users, as well as a quick acquisition capability for PPS users. The PPS's use

of two L-band frequencies, 1575.42 Mhz (L_1) and 1227.6 Mhz (L_2), provides a means for correcting for ionospheric delays [Ref. 1].

In addition to navigation, other applications are being found for the GPS, including geodesy and synchronization. Geodetic surveys usually require collecting data at a stationary site for extended periods of time. Time synchronization systems involve placing a GPS antenna at a surveyed location and solving for the receiver clock error.

Another application of receiving GPS signals at a surveyed site is for using differential corrections to improve the accuracy of a mobile unit's position solution. Using differential techniques, the accuracy available for all users can be better than five meters.

The use of interferometric techniques have proven to further improve the accuracy available, but continuous tracking of the signal carriers is required. Continuous tracking is difficult for maneuvering aircraft and is unfeasible for submersibles.

For both SPS and PPS, the accuracy of the GPS data is also dependent on the relative positions of the satellites and the user. The relationship of satellite geometry to position accuracy is explained in the next section.

B. POSITION DETERMINATION

This section develops the equations used to determine

position based on GPS measurements. For more detailed information, the reader is referred to [Ref. 1] and [Ref. 3:pp. 409-423]. The notation used in the following development is consistent with [Ref. 3:pp. 409-423].

The measurement used for the determination of a receiver's position is the GPS signal's time-of-arrival. This observable is referred to as the pseudorange, because it includes a bias due to the receiver's clock offset. The pseudorange measurement, ρ , includes the noiseless pseudorange, Ψ , along with the measurement noise, v_ρ , and time-correlated errors, β_ρ [Ref. 3:p.412].

The GPS satellites broadcast information that includes orbital parameters, referred to as ephemerides. These ephemerides are used to calculate the satellite's position, $[X_i, Y_i, Z_i]^T$ for satellite i. Four measurements are required to solve for the user's position, $\underline{x} = [x, y, z]^T$, and the receiver's clock offset, Δt . From [Ref. 3:p. 410]

$$\begin{aligned}\Psi_1 &= \sqrt{(X_1-x)^2 + (Y_1-y)^2 + (Z_1-z)^2} + c \Delta t \\ \Psi_2 &= \sqrt{(X_2-x)^2 + (Y_2-y)^2 + (Z_2-z)^2} + c \Delta t \\ \Psi_3 &= \sqrt{(X_3-x)^2 + (Y_3-y)^2 + (Z_3-z)^2} + c \Delta t \\ \Psi_4 &= \sqrt{(X_4-x)^2 + (Y_4-y)^2 + (Z_4-z)^2} + c \Delta t,\end{aligned}\tag{2.1}$$

where c is the speed of light. If the altitude is known, e.g., a vessel is on the sea's surface, than only three pseudorange measurements are required.

Most GPS receivers use Kalman filtering, which requires

the linearization of Equation 2.1. Linearizing about an approximate position, $\mathbf{x}_o = [x_o, y_o, z_o]^T$, requires the partial derivatives:

$$\begin{aligned}\frac{\delta \Psi_i}{\delta x} &= \frac{-(X_i - x_o)}{\sqrt{(X_i - x_o)^2 + (Y_i - y_o)^2 + (Z_i - z_o)^2}} \\ \frac{\delta \Psi_i}{\delta y} &= \frac{-(Y_i - y_o)}{\sqrt{(X_i - x_o)^2 + (Y_i - y_o)^2 + (Z_i - z_o)^2}} \\ \frac{\delta \Psi_i}{\delta z} &= \frac{-(Z_i - z_o)}{\sqrt{(X_i - x_o)^2 + (Y_i - y_o)^2 + (Z_i - z_o)^2}},\end{aligned}\quad (2.2)$$

for $i = 1, \dots, 4$ [Ref. 3:p. 421]. These partial derivatives are the direction cosines from the satellite to the approximated user's position.

Subtracting the predicted pseudorange, $\hat{\Psi}$, from the measured pseudorange gives the noiseless measurement equation:

$$\begin{bmatrix} \Psi_1 \\ \Psi_2 \\ \Psi_3 \\ \Psi_4 \end{bmatrix} - \begin{bmatrix} \hat{\Psi}_1(x_o) \\ \hat{\Psi}_2(x_o) \\ \hat{\Psi}_3(x_o) \\ \hat{\Psi}_4(x_o) \end{bmatrix} = \begin{bmatrix} \frac{\delta \Psi_1}{\delta x} & \frac{\delta \Psi_1}{\delta y} & \frac{\delta \Psi_1}{\delta z} & 1 \\ \frac{\delta \Psi_2}{\delta x} & \frac{\delta \Psi_2}{\delta y} & \frac{\delta \Psi_2}{\delta z} & 1 \\ \frac{\delta \Psi_3}{\delta x} & \frac{\delta \Psi_3}{\delta y} & \frac{\delta \Psi_3}{\delta z} & 1 \\ \frac{\delta \Psi_4}{\delta x} & \frac{\delta \Psi_4}{\delta y} & \frac{\delta \Psi_4}{\delta z} & 1 \end{bmatrix} \begin{bmatrix} \Delta x \\ \Delta y \\ \Delta z \\ c \Delta t \end{bmatrix}, \quad (2.3)$$

where $[\Delta x, \Delta y, \Delta z]^T = \mathbf{x} - \mathbf{x}_o$ [Ref. 3:p. 422].

These measurements are based on the direction cosines. Therefore, the accuracy of the position estimates is dependent on the satellite geometry.

C. REPRESENTATIVE GPS DATA

Data was collected with a GPS receiver to determine the statistics of position and velocity measurements. A C/A code receiver was used and therefore a comparison can be made with the published SPS position accuracies.

The specific GPS receiver used was a Magnavox model MX 4200 purchased by the Naval Postgraduate School. The data was stored on a laptop computer and software programs were written in FORTRAN to unpack the position and velocity measurements. Source code listings of these programs are provided in appendix A. A Microsoft FORTRAN compiler was used and these programs were run on IBM-compatible personal computers (XT and AT).

The GPS antenna was located on a surveyed antenna mount at Pt. Mugu, California. The surveyed position was subtracted from the measurements to obtain GPS errors. To convert from degrees of latitude and longitude to meters, correction factors were obtained from [Ref. 5].

The horizontal position errors are shown in Figure 2.1, and Figure 2.2 shows the vertical position errors. As seen in Figure 2.2, the vertical position error is constant for a period between sample 500 and sample 1000, this is due to the GPS receiver switching to an altitude-hold mode where only three satellites are used to determine the position. Figure 2.3 shows the velocity errors, which were converted from knots to meters per second.

The standard deviation of the horizontal position errors was computed to be 43.1 meters, which is close to the expected value of 50 meters based on the DOD policy. For the vertical position errors, the standard deviation was computed to be 81.7 meters, which again is close to the expected value of 78 meters. The velocity errors had a standard deviation of 0.9 meters per second.

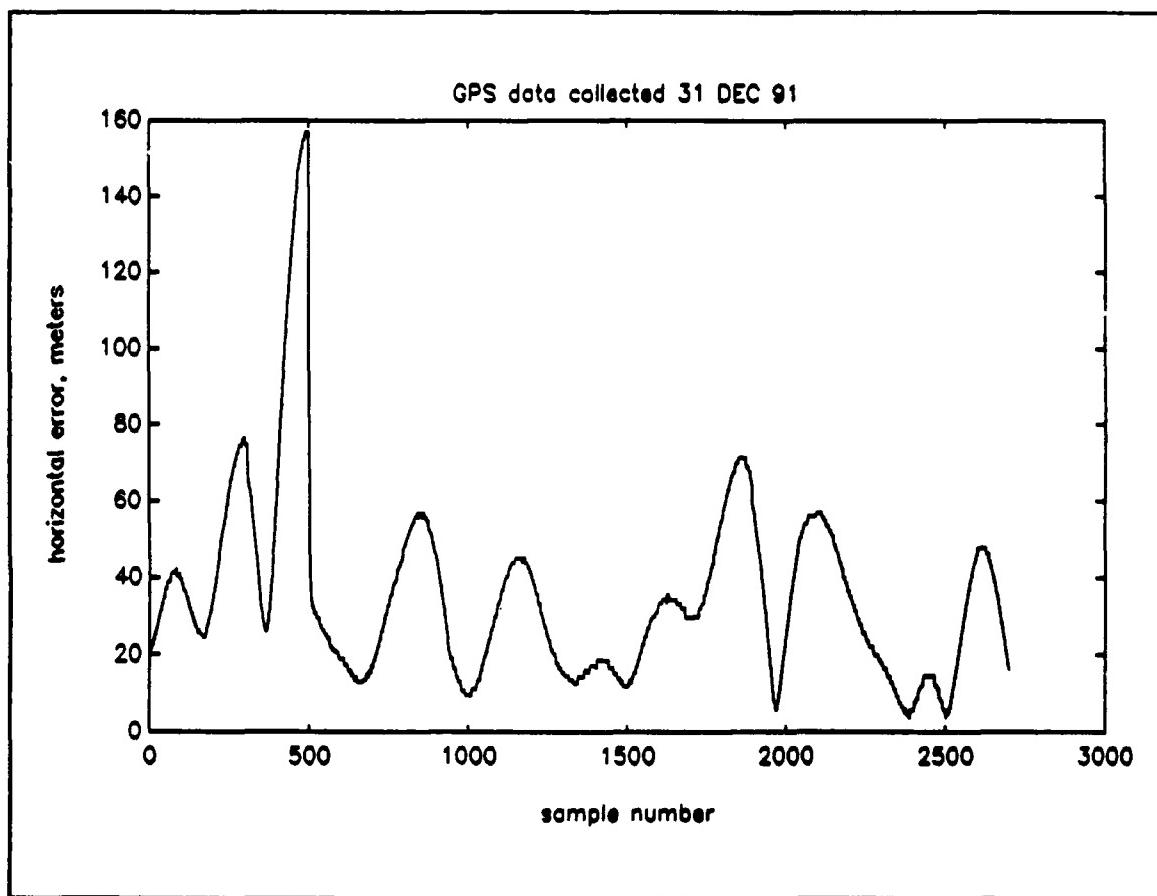


Figure 2.1: Horizontal errors in GPS position measurements.

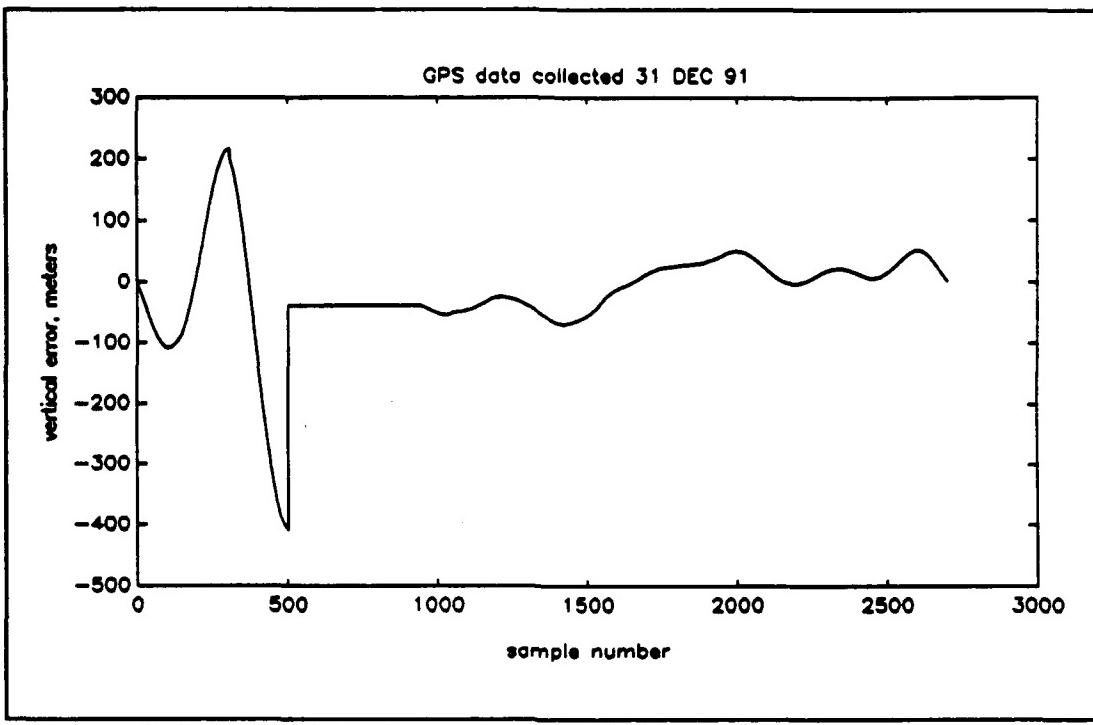


Figure 2.2: Vertical errors in GPS position measurements.

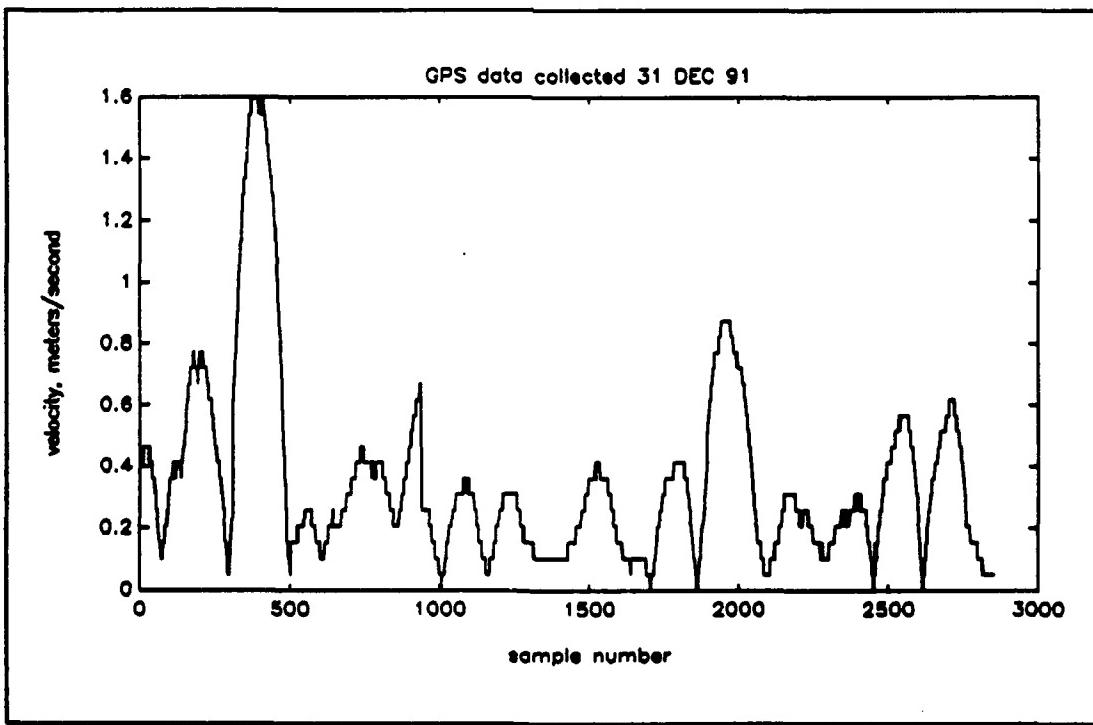


Figure 2.3: Errors in GPS velocity measurements.

III. INERTIAL NAVIGATION SYSTEMS

A. GIMBALLED SYSTEMS

An INS measures accelerations and rotations in order to compute position, velocity, and orientation of a platform, usually a moving vehicle. A typical INS uses a triad of accelerometers, mounted on orthogonal axes, to measure the specific forces, f , experienced by a vehicle. Early systems mounted these accelerometers on a platform within an arrangement of gimbals. Sets of gyroscopes and servos were used to keep the platform stable within a reference coordinate system. Any rotation of the platform was sensed by the gyros and the servos would rotate the gimbal axes as required to counter the sensed rotation. These gimballed systems are still used and provide satisfactory accuracy at the expense of size, weight, and the low reliability associated with mechanical systems.

A typical reference coordinate frame used for gimballed systems is the local-level frame ("n-frame") as illustrated in Figure 3.1. The coordinate axes are aligned with the local north, east, and down directions. The advantage of using this system is that the force of gravity is registered on only one of the accelerometers. This means that the forces sensed by the horizontal accelerometers are directly related to the vehicle accelerations in the familiar

directions of latitude, ϕ , and longitude, λ . These accelerations can be integrated once to give velocities and again to provide changes in position from some starting location.

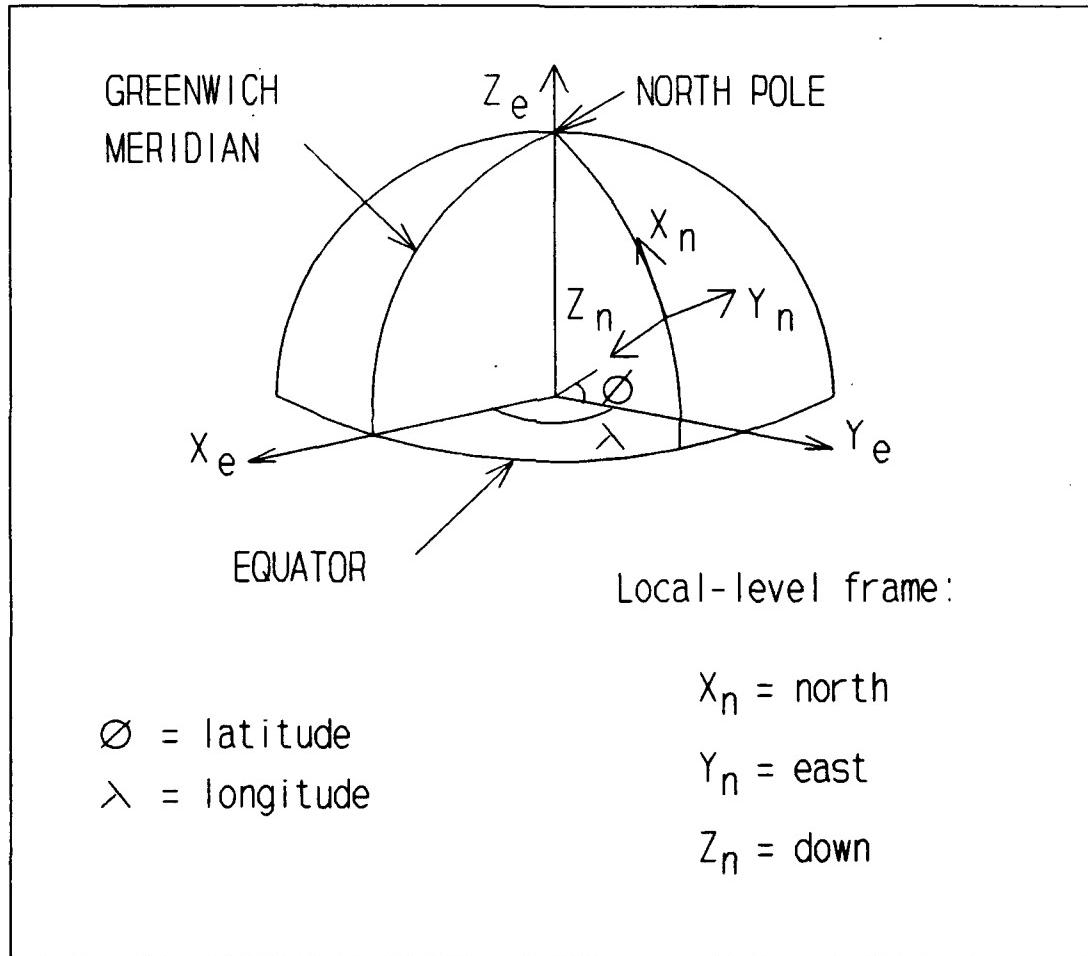


Figure 3.1: Reference frames used for inertial navigation systems. (After Ref. 6)

B. STRAPDOWN SYSTEMS

With the development of the digital computer, an alternative configuration has become practical. By hard-mounting the accelerometers to the vehicle and using gyroscopes to measure rotations, ω , a large part of the mechanical complexity can be reduced. The computer is required for keeping track of the orientation of the instrumentation package, so that the directions of the measured accelerations are known. Transformation matrices are used to convert vectors from one reference frame to another. For example, R_b^n , transforms a vector in body frame ("b-frame") coordinates to n-frame coordinates.

With these strapdown systems, the measurements are made in the b-frame, defined by the forward, right, and down directions [Ref. 7]. Thus, the rotation measurements give roll, pitch, and yaw directly. The problem is that, in addition to vehicle motion, each accelerometer measures components of other forces due to gravity and Coriolis, caused by the earth's rotation. These forces have to be estimated and subtracted, in order to obtain the accelerations due to vehicle motion.

Figure 3.2 illustrates the process of converting the inertial measurements into attitude, position (ϕ, λ, h), and velocity (v). The results are in the local-level frame if the proper transformations from the b-frame to the n-frame are included in the processing.

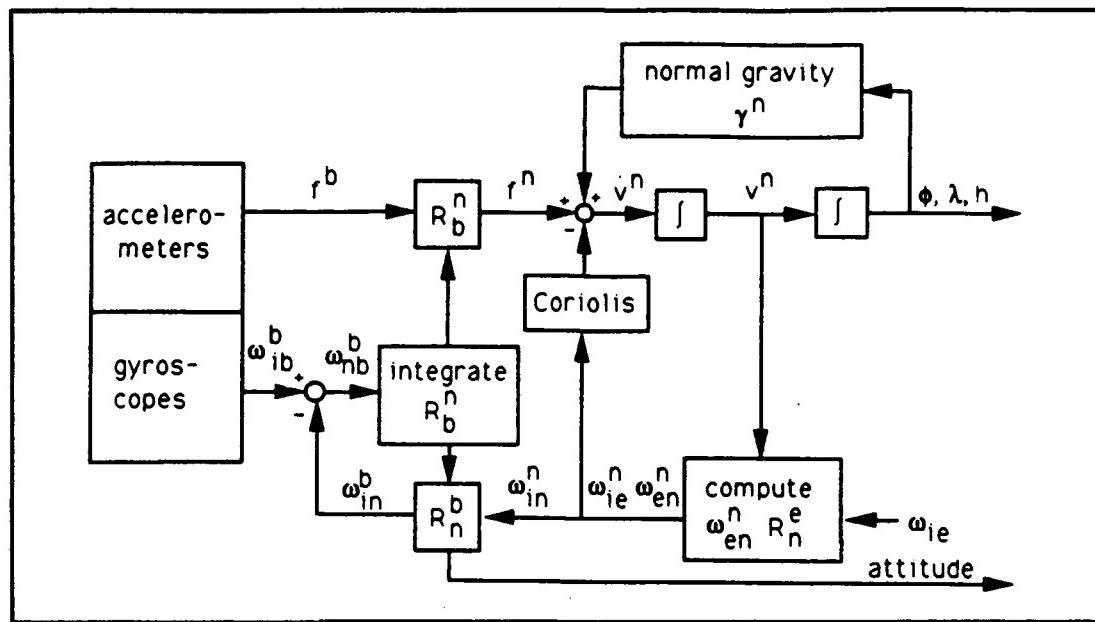


Figure 3.2: Local-level frame mechanization of INS measurement processing. (From Ref. 8)

An alternative INS mechanization, illustrated in Figure 3.3, involves transformations from the b-frame to the earth-fixed frame ("e-frame"), labeled x, y, and z in Figure 3.1.

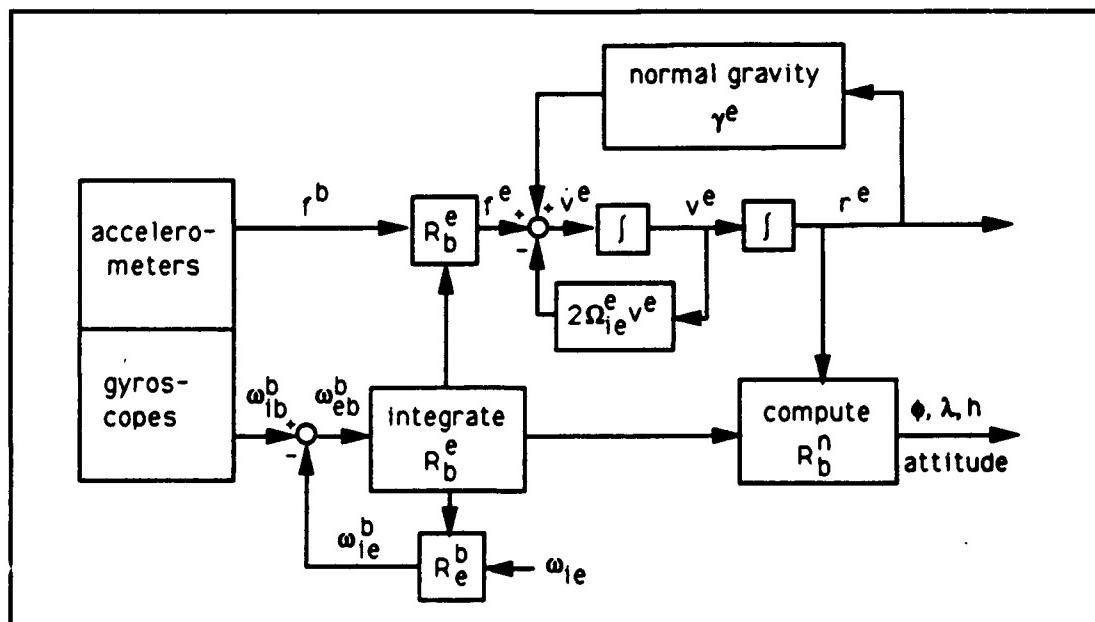


Figure 3.3: Earth-fixed frame mechanization of an INS. (From Ref. 8)

It has been shown in [Ref. 8] that the processing of measurements using the e-frame can be accomplished faster than when using the n-frame. When using the e-frame, less computing time is required for the mechanization equations and the Kalman filter. More time is required for the computation of normal gravity, but the net result is faster processing with the e-frame algorithm.

In this section we develop the equations for position and velocity estimates in the e-frame in terms of the forces acting on the vehicle. The equations are based on the developments in [Ref. 8] and [Ref. 9]. The reference frames used are the body frame (b), the earth-fixed frame (e), the local-level frame (n), and the inertial frame (i). Vectors and matrices are annotated with subscripts and superscripts depending on which frame or frames they reference. For instance, position and velocity in e-frame coordinates are denoted \underline{r}^e and \underline{v}^e respectively. Similarly, Ω_{eb}^b is the skew-symmetric matrix of ω_{eb}^b , which is the angular velocity vector of the body frame with respect to the e-frame, given in b-frame coordinates.

The skew-symmetric matrices are used to execute the cross-product operation. For instance, the Coriolis acceleration is two times the earth's rotation rate, ω_e , crossed with the vehicle's velocity,

$$2(\omega_e \hat{k}_e) \times (v^e_x \hat{i}_e + v^e_y \hat{j}_e + v^e_z \hat{k}_e) = -2\omega_e v^e_y \hat{i}_e + 2\omega_e v^e_x \hat{j}_e . \quad (3.1)$$

where the $\hat{\cdot}$ denotes a unit vector. The skew-symmetric matrix serves to simplify the notation

$$2 \Omega^e_{ie} \underline{v^e} = \begin{bmatrix} 0 & -2\omega_e & 0 \\ 2\omega_e & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} v^e_x \\ v^e_y \\ v^e_z \end{bmatrix} = \begin{bmatrix} -2\omega_e v^e_y \\ 2\omega_e v^e_x \\ 0 \end{bmatrix} , \quad (3.2)$$

where Ω^e_{ie} is the skew-symmetric matrix of the angular velocity of the earth's rotation, $\underline{\omega}_{ie}$.

The INS measurements are the specific force vector, $\underline{f^b}$, and the angular rotation rate of the body with respect to the inertial frame, $\underline{\omega}_{ib}^b$. Subtracting the earth's angular velocity vector, $\underline{\omega}_{ie}$, from $\underline{\omega}_{ib}^b$ gives

$$\underline{\omega}_{eb}^b = \underline{\omega}_{ib}^b - \underline{\omega}_{ie}^b , \quad (3.3)$$

which is used to build the skew-symmetric matrix

$$\Omega^b_{eb} = \begin{bmatrix} 0 & -(\omega_{eb}^b)_3 & (\omega_{eb}^b)_2 \\ (\omega_{eb}^b)_3 & 0 & -(\omega_{eb}^b)_1 \\ -(\omega_{eb}^b)_2 & (\omega_{eb}^b)_1 & 0 \end{bmatrix} , \quad (3.4)$$

where $(\omega_{eb}^b)_1$ is the roll rate, $(\omega_{eb}^b)_2$ is the pitch rate, and $(\omega_{eb}^b)_3$ is the yaw rate.

Accelerations are found by transforming the force measurements and subtracting the acceleration due to gravity and the Coriolis effect. The change in the transformation matrix from body to earth coordinates is given by

$$\dot{R}^e_b = R^e_b \Omega_{eb}^b . \quad (3.5)$$

The earth's angular velocity vector in body coordinates, ω_{ie}^b , is obtained from ω_{ie}^e according to

$$\omega_{ie}^b = R^b_e \omega_{ie}^e , \quad (3.6)$$

where the transformation matrix R^b_e is the inverse of R^e_b . Transformation matrices involved are orthogonal, and therefore their inverses are equal to their transposes:

$$R^b_e = [R^e_b]^{-1} = [R^e_b]^T . \quad (3.7)$$

We can relate different transformation matrices as [Ref. 6]:

$$R^e_b = R^n_b R^n_e . \quad (3.8)$$

where the transformation matrix, R^n_b , is obtained from initialization and the time history of body rotations.

Using the geometry of Figure 3.1, and unit vectors along the e-frame and n-frame coordinate axes:

$$\begin{bmatrix} \hat{i}_n \\ \hat{j}_n \\ \hat{k}_n \end{bmatrix} = R^{\text{e}_n} \begin{bmatrix} \hat{i}_{\text{e}} \\ \hat{j}_{\text{e}} \\ \hat{k}_{\text{e}} \end{bmatrix} = \begin{bmatrix} -\sin\phi \cos\lambda \hat{i}_{\text{e}} - \sin\phi \sin\lambda \hat{j}_{\text{e}} + \cos\phi \hat{k}_{\text{e}} \\ -\sin\lambda \hat{i}_{\text{e}} + \cos\lambda \hat{j}_{\text{e}} \\ -\cos\phi \cos\lambda \hat{i}_{\text{e}} - \cos\phi \sin\lambda \hat{j}_{\text{e}} - \sin\phi \hat{k}_{\text{e}} \end{bmatrix}, \quad (3.9)$$

where ϕ denotes latitude and λ denotes longitude.

Therefore,

$$R^{\text{e}_n} = \begin{bmatrix} -\sin\phi \cos\lambda & -\sin\phi \sin\lambda & \cos\phi \\ -\sin\lambda & \cos\lambda & 0 \\ -\cos\phi \cos\lambda & -\cos\phi \sin\lambda & -\sin\phi \end{bmatrix} \quad (3.10)$$

and

$$R^{\text{e}_b} = [R^{\text{e}_n}]^T = \begin{bmatrix} -\sin\phi \cos\lambda & -\sin\lambda & -\cos\phi \cos\lambda \\ -\sin\phi \sin\lambda & \cos\lambda & -\cos\phi \sin\lambda \\ \cos\phi & 0 & -\sin\phi \end{bmatrix}. \quad (3.11)$$

After initialization, R^{e_b} , is obtained by integrating

$$\dot{R}^{\text{e}_b} = R^{\text{e}_b} \Omega^b_{eb}, \text{ i.e.,}$$

$$R^{\text{e}_b}(k) = R^{\text{e}_b}(0) + \sum_0^k R^{\text{e}_b} \Omega^b_{eb} (\Delta t), \quad (3.12)$$

where k is the number of the current sample and Δt is the sampling interval.

In summary, the mechanization equation is given by

$$\begin{bmatrix} \dot{x}^{\text{e}} \\ \dot{y}^{\text{e}} \\ \dot{z}^{\text{e}} \end{bmatrix} = \begin{bmatrix} \underline{v}^{\text{e}} \\ R^{\text{e}_b} \underline{f}^b - 2 \Omega^b_{ie} \underline{v}^{\text{e}} + \underline{g}^{\text{e}} \\ R^{\text{e}_b} \Omega^b_{eb} \end{bmatrix}, \quad (3.13)$$

where \underline{g}^{e} is the gravity vector.

C. GYROSCOPE PERFORMANCE CHARACTERISTICS

The performance characteristics used for comparing gyros include gyro drift, scale factor error, and random walk noise. Other parameters of importance are the size, weight, cost, and dynamic range. The dynamic range is the range of input rotation rates that can be measured correctly.

The gyro drift is also referred to as bias error or bias stability and is given in units of angle per unit of time (e.g., deg/sec). It is also common to describe a gyro's performance in terms of nautical miles (nmi) per hour. This measure is based on how the gyro would perform in an INS. Figure 3.4 shows how the position error grows with time for a given gyro drift. This plot shows that for a gyro bias of 0.01 deg/hr, in combination with an accelerometer with a bias of 0.0001 m/s^2 , the performance is on the order of 1 nmi/hr.

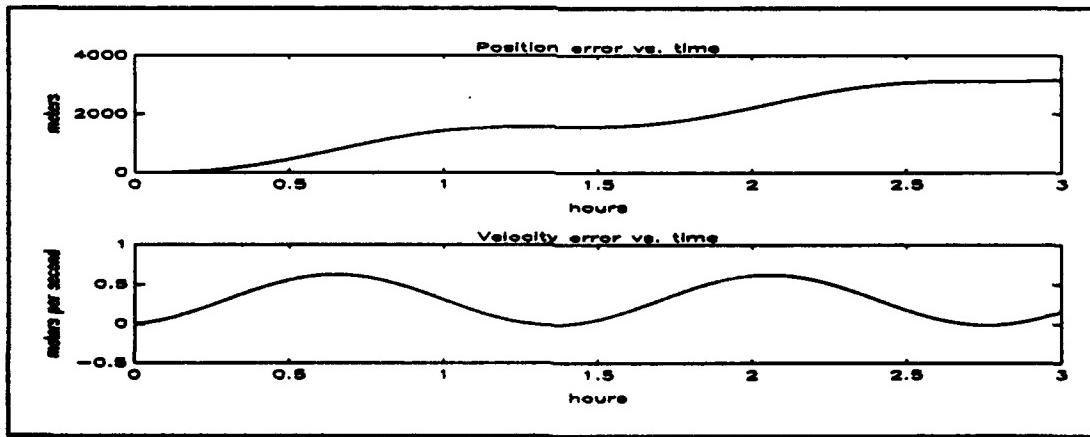


Figure 3.4: Errors in an INS with a gyro drift of 0.01 deg/hr and an accelerometer bias of 0.0001 m/s^2 .

Figure 3.4 is based on a model given in [Ref. 10]. This model is valid for mission times of less than three hours. For missions of this duration, the Schuler oscillation is important, but the 24 hour effects can be neglected. The model is given by

$$\begin{aligned} p(t) &= \frac{a}{\omega_s^2} (1 - \cos \omega_s t) + gR_e (t - \frac{1}{\omega_s} \sin \omega_s t) \\ v(t) &= \frac{a}{\omega_s} \sin \omega_s t + gR_e (1 - \cos \omega_s t) \end{aligned}, \quad (3.14)$$

where $p(t)$ is the position (given as a function of time), $v(t)$ is the velocity as a function of time, a is the accelerometer bias, g is the gyro drift, R_e is the earth's radius, and ω_s is the frequency of the Schuler oscillation.

The scale factor is the resolution of the system. The scale factor error is given in parts per million (ppm) or percent (%). Gyroscope noise is primarily due to random walk and is given in units of degrees per square root hour.

System requirements are different depending on the application. A submarine for launching ballistic missiles can carry a large instrument package and requires high accuracy navigation. Since submarines remain submerged for extended periods of time, they cannot tolerate large error growth. Fortunately, these characteristics are compatible, i.e., for higher accuracy and smaller error growth, a large volume is required.

For commercial aircraft a medium-grade system is sufficient and usually preferred, due to the reduced cost. Such a system, with medium-level accuracy, is commonly referred to as an Attitude Heading Reference System (AHRS). For a tactical missile with a short flight time, a low accuracy instrument is acceptable.

Very high accuracy gyroscopes are used for scientific experiments related to geophysics and relativity. Geodesists and surveyors routinely use INS's, including those with Ring Laser Gyros (RLG's). Table 3.1 summarizes the main performance characteristics and provides typical value ranges for gyros of different levels of accuracy.

Table 3.1: FAMILY OF GYROSCOPE APPLICATIONS (After Ref. 11)

PERFORMANCE PARAMETER	LOW-GRADE (munitions)	MEDIUM-GRADE (missiles)	HIGH-GRADE (submarines)
Bias Stability	0.5 - 1 deg/sec	0.1 - 1 deg/hr	0.001 - 0.01 deg/hr
Random Walk (deg/root hr)	0.1 - 1	0.02 - 0.25	0.001
Scale Factor Linearity	0.1 - 5 %	100 - 5000 ppm	< 10 ppm

D. OPTICAL GYROSCOPES

1. Advantages of Optical Gyroscopes

Prior to the development of the laser, mechanical gyroscopes were used extensively. These mechanical gyros consist of a spinning mass along with a motor to provide the

torque required to keep the mass spinning. Great care is taken during manufacturing to produce a balanced mass and reduce friction, but there is always some residual imbalance and friction contributing to measurement error. These mechanical devices are sensitive to large accelerations (g 's) and have larger errors in a high-dynamic environment.

The idea of using interferometry to measure rotation rates dates back to before the turn of the century, and Sagnac demonstrated this capability in 1913 during ether experiments [Ref. 12]. The first experiments with a laser gyro were performed in 1962 [Ref. 12]. Since that time the Ring Laser Gyro (RLG) has been developed into a practical instrument and is operational on aircraft ranging from the commercial Boeing 757/767 series [Ref. 13] to the military F-15 and UH-60 [Ref. 14]. Research and development of an operational Fiber Optic Gyro (FOG) is ongoing and flight tests have been conducted [Ref. 14].

The major advantage of optical gyroscopes is the large reduction in the number of moving parts. In fact, FOG's are completely solid-state [Ref. 15]. This primary advantage provides several secondary benefits. With fewer moving parts, optical gyroscopes are easier to maintain, are more reliable, and are more rugged. In addition, for a given level of accuracy, optical gyros can be smaller and lighter, and have a lower total parts count. Components for FOG's are readily available due to the increased use of

fiber optics in communication systems. All of these factors contribute to lower costs.

Additional advantages are the large reductions in both warm-up time and sensitivity to high-dynamic maneuvers. According to [Ref. 15], FOG's are being developed for "smart munitions" which experience up to 20,000 g's when fired from a cannon.

Optical gyros can be used to support navigation of platforms ranging from space vehicles to submarines. Indeed, this technology is currently replacing the traditional mechanical implementation.

2. Theory of Operation: Sagnac Effect

The Sagnac effect can be described [Ref. 16] in terms of the transit times of counter-propagating beams traveling in a circular optical cavity of radius R, as illustrated in Figure 3.5. The light enters the system from point A and the beam splitter diverts some of the light into the clockwise path and some of the light into the counter-clockwise path. If there is no rotation, then the beam splitter will not move and the two beams will recombine and return to point A. In this case, the transit time τ is the same in both directions and is given by

$$\tau = \frac{2\pi R}{c} . \quad (3.15)$$

SAGNAC EFFECT

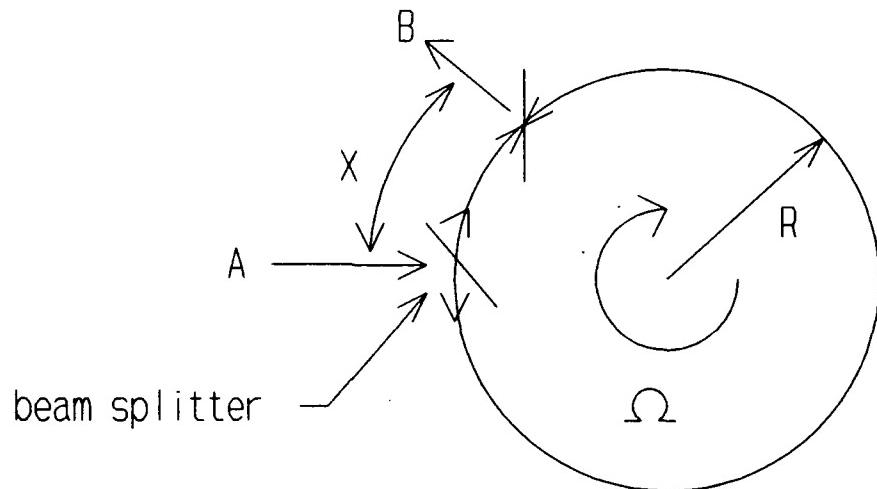


Figure 3.5: Illustration of the Sagnac effect.

When the sensor rotates at a rate of Ω rad/sec, the beam splitter moves a distance X from point A to point B, where

$$X = R\Omega\tau \quad . \quad (3.16)$$

For the beam traveling in the direction of the rotation, the transit time is

$$\tau_+ = \frac{2\pi R + X}{c} = \frac{2\pi R + R\Omega\tau}{c} \quad . \quad (3.17)$$

For the oppositely directed beam, the transit time is

$$\tau_- = \frac{2\pi R - R\Omega\tau}{c} \quad . \quad (3.18)$$

Factoring out the transit time and combining the two equations:

$$\tau_s \left(1 \mp \frac{R\Omega}{c} \right) = \frac{2\pi R}{c} . \quad (3.19)$$

The difference in transit times is

$$\Delta\tau = \tau_+ - \tau_- = \frac{2\pi R}{c-R\Omega} - \frac{2\pi R}{c+R\Omega} . \quad (3.20)$$

Placing the two terms over a common denominator gives

$$\Delta\tau = \frac{4\pi R^2 \Omega}{c^2 - R^2 \Omega^2} . \quad (3.21)$$

Since $c^2 > R^2 \Omega^2$:

$$\Delta\tau = \frac{4\pi R^2 \Omega}{c^2} . \quad (3.22)$$

The optical path length difference is

$$\Delta L = c \Delta\tau = \frac{4\pi R^2 \Omega}{c} . \quad (3.23)$$

For a circle of area $A = \pi R^2$:

$$\Delta L = \frac{4A\Omega}{c} . \quad (3.24)$$

It can be shown that Equation 3.24 is valid for any geometric shape of area A. This equation shows how a rotation causes a change in optical path length. It also shows that increasing the area of the optical path will

increase the optical path length difference. For rotation rates of interest in most applications, an impractical area would be required if no alternatives were available.

Fortunately, by using N turns of a fiber optic cable, the area is multiplied by N and practical systems can be built.

The FOG is a passive interferometer and using a coil of N turns of cable provides the required sensitivity. A RLG is an active interferometer and its ability to provide the necessary sensitivity is described in the next section.

3. Ring Laser Gyros

An active interferometer improves the sensitivity because the laser frequency depends on the cavity length [Ref. 16]. With a gain medium within the optical cavity, the condition for oscillation is that an integral number of wavelengths fits within the cavity length. In a ring laser the two oppositely traveling waves can have different amplitudes and frequencies. A small change in the optical path length leads to a small change in frequency given by

$$\frac{\Delta v}{v} = \frac{\Delta L}{L} . \quad (3.25)$$

Since the optical frequency, v , is on the order of 10^{14} Hz, small differences in length lead to changes in frequency that are large enough to be measured. From the Sagnac effect,

$$\Delta L = \frac{4A\Omega}{C} , \quad (3.26)$$

and the definition of optical frequency ($v=c/\lambda$), the frequency difference is given by

$$\frac{\Delta v}{v} = \frac{\Delta v}{\left(\frac{c}{\lambda}\right)} = \frac{\Delta L}{L} = \frac{\left(\frac{4A\Omega}{C}\right)}{L} . \quad (3.27)$$

Therefore, the beat frequency is:

$$\Delta v = \frac{4A\Omega}{CL} \left(\frac{C}{\lambda}\right) = \frac{4A\Omega}{L\lambda} . \quad (3.28)$$

The net number of accumulated counts is found by integration:

$$N = \int_0^t \Delta v dt = \int_0^t \frac{4A\Omega}{L\lambda} dt = \frac{4A}{L\lambda} \int_0^t \Omega dt = \frac{4A\theta}{L\lambda} , \quad (3.29)$$

where θ is the net angle through which the gyro has turned.

For example: given $L = 39.6$ cm, $A = 75.45$ cm^2 , and a He-Ne laser with a wavelength of 0.633 μm , setting $N = 1$ gives a resolution (scale factor) of $\theta_{\min} = 8.3 \times 10^{-6}$ rad = 1.7 arcsec. This is equivalent to 7.6×10^5 counts per revolution or 2118 counts/deg.

The read-out is obtained by using a prism behind one of the RLG's mirrors. The intensity of the resulting fringe pattern is given by

$$I = I_o [1 + \cos(2\pi\varepsilon/\lambda + \Delta\omega t + \phi)] , \quad (3.30)$$

where $\Delta\omega$ is the angular beat frequency, λ/ε is the fringe

spacing, x is the distance along the pattern, and ϕ is an arbitrary phase angle [Ref. 16]. The fringes can be counted by using detectors smaller than the fringe spacing. Two detectors, separated by a quarter fringe, can be used to sense the direction of rotation.

For an ideal RLG, the output N is a linear function of the input θ . The relationship between the input and output is referred to as the characteristic curve. Any deviation from the ideal case is considered a scale-factor error. The scale factor determines the slope of the characteristic curve. A phenomenon, referred to as mode pulling, changes the slope of the characteristic curve. Mode pulling is due to changes in the index of refraction within the gain medium.

Another problem in RLG's is the lock-in due to back-scattering from the mirrors. At low rotation rates, a RLG will not respond. The typical method of overcoming lock-in is to mechanically dither the RLG, or its mirrors, using piezo-electric transducers so that the RLG operates outside of the problem area. The characteristic curve can also be shifted up or down so that a non-zero reading is obtained without any rotation. This effect is called null shift. Anything besides a rotation that influences the oppositely directed beams in an unequal manner will contribute to the null-shift effect. Such phenomena are known as nonreciprocal effects.

In order to get three RLG's with the largest area possible for a given volume, the three optical cavities can be combined within a single block [Ref. 16]. With this configuration, each mirror is shared by two of the optical paths, thus reducing the parts count [Ref. 17].

IV. INS ERROR ESTIMATION

A. INS ERROR MODEL

The following development uses the techniques in [Ref. 8] and [Ref. 18] to derive the equations governing the INS error dynamics. This INS error model is used in the Kalman filter to estimate the INS errors. There are 15 states based on five vectors. The first three states are the e-frame components of the attitude error, $\underline{\alpha}$. The next six states are the e-frame components of the position error, Δp , and the velocity error, Δv . The last two vectors are the gyro drift, d , and the accelerometer bias, b , given in b-frame coordinates.

The state vector has been augmented with the drift and bias terms, because they are modeled as Gauss-Markov processes. This augmentation has been done so that the system noise, w is white. The gyro drift noise and the accelerometer noise will have standard deviations of σ_d , and σ_b , respectively. The resulting state vector [Ref. 8] for the error signal can then be written as

$$\underline{x} = (\underline{\alpha}, \Delta p, \Delta v, d, b)^T \quad (4.1)$$

and it is modeled as a standard state space equation

$$\dot{x} = [F] x + w, \quad (4.2)$$

where $[F]$ is the dynamics matrix, determined on the basis of the system error equations.

In particular, the misalignments are modeled as

$$\dot{\theta} = -\Omega^e_{ie} \theta + R^e_b d . \quad (4.3)$$

The position errors are only dependent on the velocity errors:

$$\dot{\delta p} = [I] \dot{\delta v}, \quad (4.4)$$

where $[I]$ is the identity matrix.

The velocity errors can be affected by the normal gravity error. However the height of the AUV at the time of the measurement will be known to be sea level. Therefore, according to [Ref. 18], this error source can be neglected. We are left with the effects of the attitude errors coupled into the force measurements, the velocity errors coupled into the Coriolis force calculations, and the accelerometer biases. This is expressed by the equation

$$\dot{\delta v} = -F^e \theta - 2 \Omega^e_{ie} \delta v + R^e_b b , \quad (4.5)$$

where F^e is the skew-symmetric matrix of the measured forces, f^b , transformed into earth-fixed coordinates.

Modeling the gyro drifts and the accelerometer biases as first-order Gauss-Markov processes with $\zeta = 1$ / (drift correlation time) and $\beta = 1$ / (bias correlation time):

$$\dot{d} = -\zeta d + w, \quad (4.6)$$

and

$$\dot{b} = -\beta b + v. \quad (4.7)$$

The resulting model for the system errors then yields a dynamics matrix

$$[F] = \begin{bmatrix} -\Omega^e_{ie} & [0] & [0] & R^e_b & [0] \\ [0] & [0] & [I] & [0] & [0] \\ -F^e & [0] & -2\Omega^e_{ie} & [0] & R^e_b \\ [0] & [0] & [0] & -\zeta[I] & [0] \\ [0] & [0] & [0] & [0] & -\beta[I] \end{bmatrix}, \quad (4.8)$$

where $[0]$ is a three-by-three null matrix.

B. KALMAN FILTER EQUATIONS

Kalman filtering techniques can be used to estimate the INS errors based on the model given by Equations 4.1, 4.2, and 4.8. For the given AUV scenario, Equation 4.2 can be rewritten as

$$\dot{x} = [F] x + G w, \quad (4.9)$$

where G is the system noise matrix, and, from Equations 4.3 through 4.7,

$$G = \begin{bmatrix} [0] & [0] \\ [0] & [0] \\ [0] & [0] \\ [I] \sigma_d & [0] \\ [0] & [I] \sigma_b \end{bmatrix} . \quad (4.10)$$

Standard techniques, such as those described in [Ref. 3], can be used to discretize Equation 4.9. The resulting discrete version is

$$\underline{x}(k+1) = \phi(k) \underline{x}(k) + \Gamma(k) \underline{w}(k) , \quad (4.11)$$

where $\Gamma(k)$ is the discrete version of G , and $\underline{w}(k)$ is the white system noise vector. The transition matrix, $\phi(k)$, relates the state vector at time t_k , $\underline{x}(k)$, to the state vector at the next sampling time, $\underline{x}(k+1)$, under noiseless conditions.

Estimates of the state vector, $\hat{\underline{x}}(k)$, are obtained by filtering the measurements, $\underline{z}(k)$. At each sample time, t_k , a new measurement is taken, which can be represented by

$$\underline{z}(k) = H(k) \underline{x}(k) + \underline{y}(k) , \quad (4.12)$$

where $H(k)$ is the observation matrix, and $\underline{y}(k)$ is the measurement noise.

For GPS-aiding, the measurements are the errors in attitude, position, and velocity. Therefore,

$$H = \begin{bmatrix} [I] & [0] & [0] & [0] & [0] \\ [0] & [I] & [0] & [0] & [0] \\ [0] & [0] & [I] & [0] & [0] \\ [0] & [0] & [0] & [0] & [0] \\ [0] & [0] & [0] & [0] & [0] \end{bmatrix}, \quad (4.13)$$

and $\underline{v}(k)$ consists of the noise on the aiding systems, including GPS, the depth meter, and the compass.

In order to use a Kalman filter, the noise processes, $\underline{w}(k)$ and $\underline{v}(k)$, must be uncorrelated. In addition, the non-zero, diagonal components of the system and measurement noise covariance matrices, $Q(k)$ and $R(k)$, must be known. For AUV applications, these conditions are met, but the covariances differ depending on the conditions and equipment used. The best accuracy is obtained with a high-grade INS aided by differential GPS.

The recursive Kalman filter algorithm is [Ref. 3:p. 235]

$$\begin{aligned} K(k) &= P^-(k) H^T [H P^-(k) H^T + R(k)]^{-1} \\ \hat{x}(k) &= \hat{x}^-(k) + K(k) [z(k) - H \hat{x}^-(k)] \\ P(k) &= [[I] - K(k) H] P^-(k) \\ P^-(k+1) &= \phi(k) P(k) \phi^T(k) + Q(k) \\ \hat{x}^-(k+1) &= \phi(k) \hat{x}(k), \end{aligned} \quad (4.14)$$

where the minus sign superscript refers to conditions just prior to incorporating the measurement. The $K(k)$ matrices are the Kalman gains and the $P(k)$ matrices are the error covariance matrices, based on the estimation errors given by

$$\hat{x}(k) = x(k) - \hat{x}(k). \quad (4.15)$$

This algorithm requires an initial estimate of $P(0)$ and $\hat{x}(0)$. Since the INS errors are zero mean, the initial estimates of the state variables, $\hat{x}(0)$, are zeros. The values used for the initial variances in the error covariance matrix, $P(0)$, are based on a commercially available INS. This particular INS was designed for civilian aircraft and measures position with an accuracy of two nautical miles per hour ($2\sigma = 2$ nmi/h) [Ref. 18:p.24]. The initial variances are listed in Table 4.1.

Table 4.1: INITIAL VARIANCES IN THE ERROR COVARIANCE MATRIX
(After Ref. 18:pp. 63-64)

STATE VARIABLE	INITIAL ERROR VARIANCE
\hat{z}	2.35×10^{-7} (radians) 2
$\hat{\delta p}$	400 (meters) 2
$\hat{\delta v}$	10^{-8} (meters/second) 2
\hat{d}	2.35×10^{-15} (radians/second) 2
\hat{b}	10^{-8} (meters/second 2) 2

V. COMPUTER SIMULATION

A computer simulation has been conducted to verify the theory explained in the preceding sections. As illustrated in Figure 5.1, this simulation included three main parts: an AUV model, an INS error model, and a Kalman filter. The simulation software includes sections that perform additional functions, such as providing control inputs to the AUV model, and adding noise to the measurements. This software was developed using the MATLAB applications package.

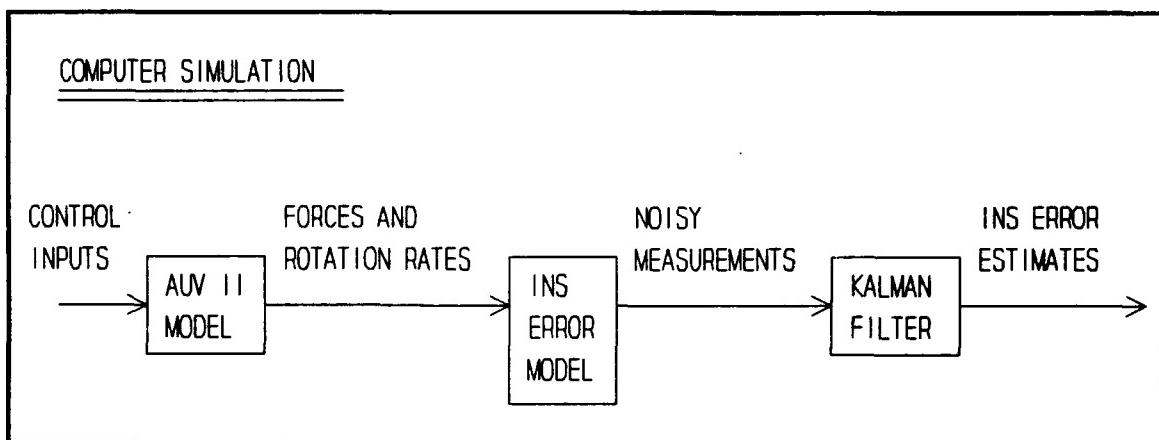


Figure 5.1: Block diagram of computer simulation.

An existing, nonlinear AUV model [Ref. 19:pp. 18-34] was used to create realistic sequences of forces and rotations. This model, AUV2, was written in the C software language and compiled for use with MATLAB. The AUV2 program is based on the parameters of the Naval Postgraduate School's second AUV.

The flight profile for the AUV consisted of an initial surface interval followed by a series of dives. The initial surface interval was a two-minute period during which the AUV remained near the surface for continuous GPS measurements. This was done to allow the Kalman filter to settle down. A reference depth of one half meter was used because the AUV2 model is only valid when the vehicle is submerged. The hydrodynamic equations are different for a vehicle on the surface. The reference depth was used in a feedback loop, along with a gain term, to set the angle of the AUV's bow and stern planes. Several trial runs were conducted to find a suitable value for the gain.

Following the initial surface interval, the AUV model was given a sinusoidal input to simulate thirty dives. Each dive lasted for two minutes, and a GPS measurement was made at the end of each dive.

The forces and rotations output from AUV2 are used by an INS error model to create noisy measurements. This error model is based on the equations in Chapter 4 and is illustrated in Figure 5.2.

To initialize the transformation matrix, R_e^s , it is assumed that the AUV is aligned with the local-level frame. Thus the transformation matrix from body to local-level coordinates, R_b^s , is reduced to the identity matrix, [I], and, from Equation 3.7, we have $R_b^s = R_e^s$. A latitude and

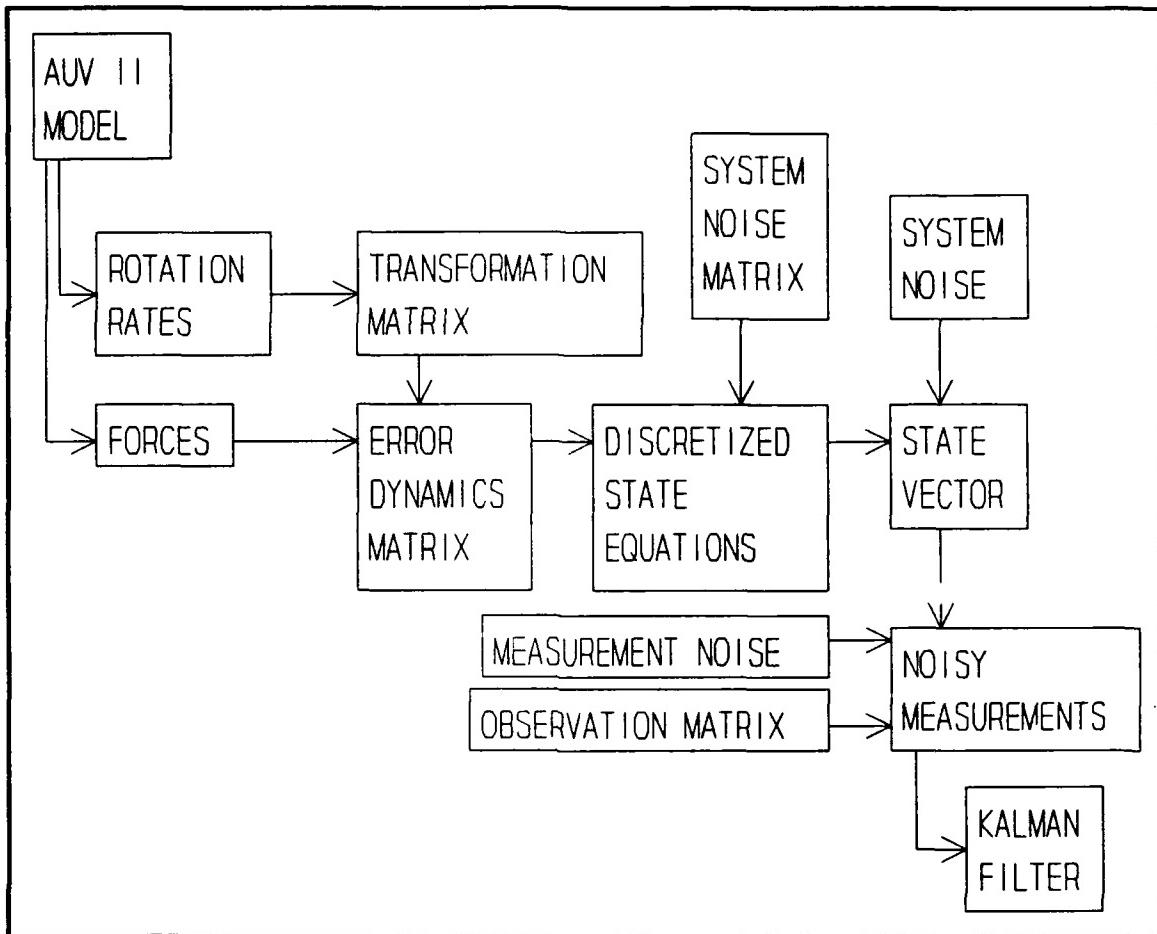


Figure 5.2: Block diagram of INS error model.

longitude near Monterey, California was used to initialize R_b^e in accordance with Equation 2.10. Equation 2.11, based on the body rotations, is used to update R_b^e .

The forces are used, along with R_b^e , to update the error dynamics matrix, $[F]$. The system noise matrix, G , is then used, along with $[F]$, to form the discrete-time version of the state equation. To convert Equation 4.9 into Equation 4.11, the following approximations are useful [Ref. 3:p.224] and [Ref. 20]:

$$\begin{aligned}\Phi &= [I] + F \Delta t + \frac{F^2(\Delta t)^2}{2!} + \dots \\ \Gamma &= [[I](\Delta t) + \frac{F(\Delta t)^2}{2!} + \frac{F^2(\Delta t)^3}{3!} + \dots]G.\end{aligned}\quad (5.1)$$

The MATLAB command "c2d" also converts the state equation from the continuous to the discrete version. A comparison was done that showed that the approximating equations ran faster than the "c2d" algorithm. Several runs were executed to find out how many terms were required in Equations 5.1 to reduce the approximation errors to negligible values.

Using MATLAB's random number generator, and specifying a normal distribution with a given standard deviation, system noise is added to create a simulated state vector. Similarly, measurement noise is added and the observation matrix is used to create a simulated measurement vector. These noisy measurements are then fed into the Kalman filter to estimate the simulated state vector.

The values used for the standard deviations of the system noise were 0.01 deg/h for gyro drifts and 10 mgal for accelerometer biases [Ref. 18:p. 64]. These values were based on an INS that uses RLG's. Converting the units gives

$$\begin{aligned}\sigma_d &= \left(\frac{0.01^\circ}{hour} \right) \left(\frac{\pi}{180^\circ} \right) \left(\frac{1hour}{3600s} \right) = 5 \times 10^{-8} rad/s \\ \sigma_b &= (10 \times 10^{-3} gal) \left(\frac{0.01m/s^2}{gal} \right) = 10^{-4} m/s^2.\end{aligned}\quad (5.2)$$

For the measurement noise, the standard deviations were based on GPS statistics for position and velocity, and two levels of accuracy for attitude measurements. For differential GPS, the standard deviations of position measurement noise, σ_p , is 2.5 meters, and the standard deviation of velocity measurement noise, σ_v , is 0.05 meters per second. For the C/A code GPS simulation, $\sigma_p = 50$ m, and $\sigma_v = 0.9$ m/s.

A standard deviation of 10^{-6} radians was used to simulate a high accuracy attitude measurement. A low accuracy simulation was run using a value of 0.1 radian.

These standard deviations were squared to obtain the variances required for the measurement noise covariance matrix, R. Table 5.1 lists the system noise variances.

Table 5.1: STATE VARIABLE VARIANCES (After Ref. 18:p. 65)

STATE VARIABLE	SYSTEM NOISE VARIANCE
attitude error	2.35×10^{-12} (radians) ²
position error	0 (meters) ²
velocity error	10^{-6} (meters/second) ²
gyro drift	3×10^{-12} (radians/second) ²
accelerometer bias	3.7×10^{-7} (meters/second ²) ²

A correlation time of 40 hours was used for the Gauss-Markov process, i.e., gyro drifts and accelerometer biases [Ref. 18:p. 65].

Several MATLAB command files ("m-files") were generated to perform the separate tasks of the computer simulation.

These files are included in Appendix B. The file "auv2surf.m" calls the AUV2 model with control inputs to generate vehicle state outputs for the surface interval. The control inputs are bow and stern plane angle commands based on the difference between a reference depth of 0.5 meters and the feedback depth output from AUV2. The file "auv2dive.m" calls AUV2 with a sinusoidal input to simulate a series of dives. The file "plotdpth.m" graphs the depth during the first five dives.

The files "sim1surf.m", "sim2surf.m", "sim3surf.m", and "sim4surf.m" use AUV2 output states during the surface interval to simulate the INS errors and to generate error estimates with the Kalman filter. The only difference in these four files are the values used for the standard deviations of position, velocity, and attitude measurement noise. The file "plot1.m" graphs the results from the simulated surface interval runs.

The file "simdives.m" uses the results from the surface runs and "auv2dive.m" to simulate the series of dives. The file "plot2.m" graphs the results of the dive runs. The file "esterr.m" calculates and plots the INS error estimate errors following each simulated series of dives.

After the simulation software was debugged, and observability was verified, four runs were executed to compare the combinations of differential or C/A code GPS with high-accuracy or low-accuracy attitude measurements.

Table 5.2 lists the combinations used for the simulations. The results of the four simulation runs are presented and discussed in the next chapter.

Table 5.2: ACCURACY LEVELS DURING THE COMPUTER SIMULATIONS

RUN #	GPS	ATTITUDE MEASUREMENTS
1	differential	high accuracy
2	differential	low accuracy
3	C/A code	high accuracy
4	C/A code	low accuracy

VI. RESULTS

The results from the four simulation runs have been graphed using MATLAB plotting commands. The AUV depth during the initial surface interval and the following series of dives were the same for all four runs. Figure 6.1 shows the depth during the surface interval. Figure 6.2 shows the depth during the first five dives.

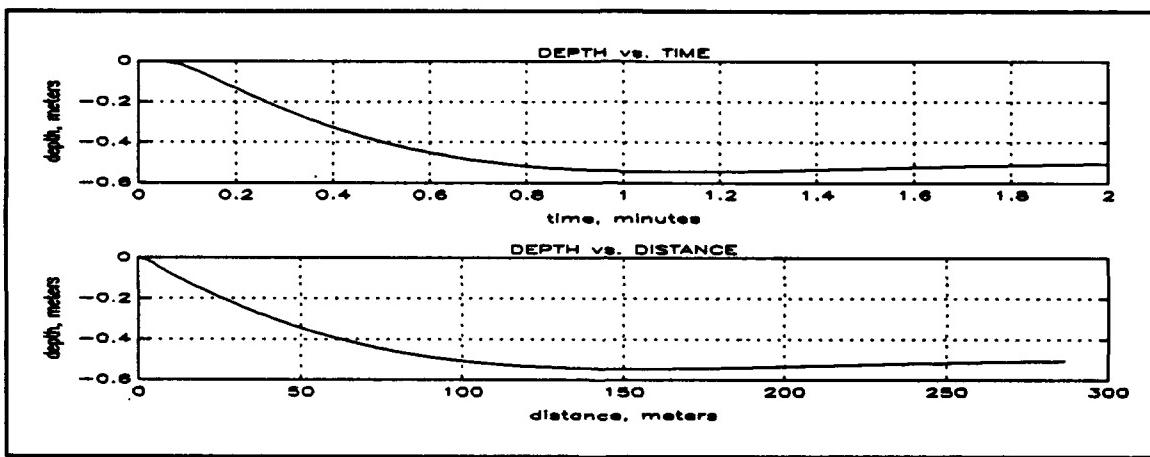


Figure 6.1: AUV depth during initial surface interval.

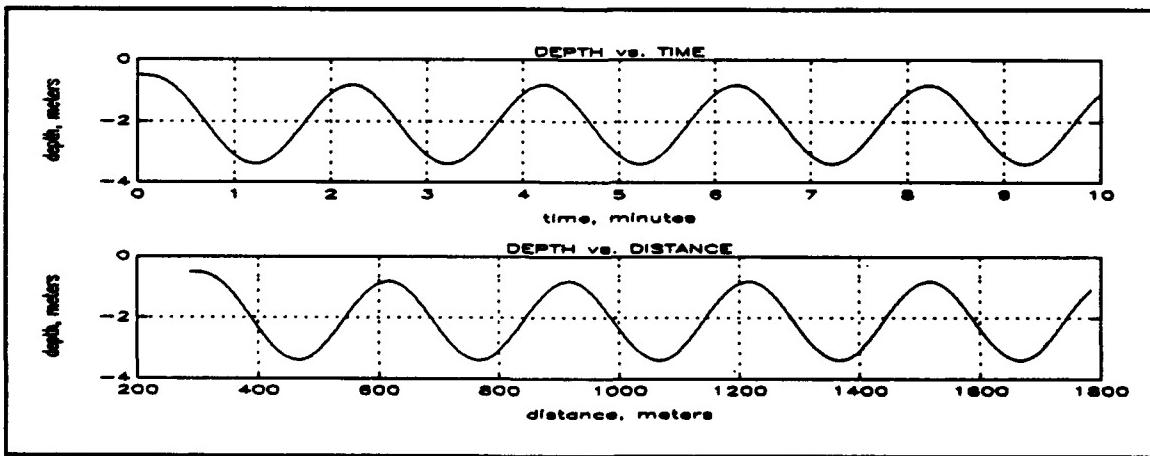


Figure 6.2: AUV depth during first five dives.

Figures 6.3 through 6.17 show the results from the first run. Figures 6.18 through 6.62 show the results from the next three runs. The first five figures from each run show the output states during the surface interval. The next five figures show the output states during the dive series. These output states include the modelled INS errors and the INS error estimates from the Kalman filter. The dashed curves are the error estimates. The last five figures in each set show the error estimate errors.

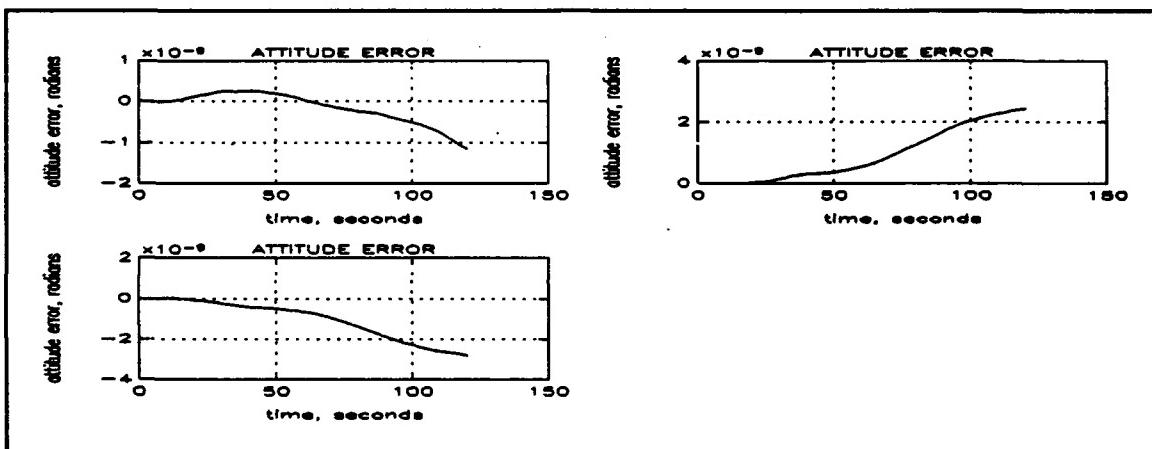


Figure 6.3: First-run, surface-interval attitude error.

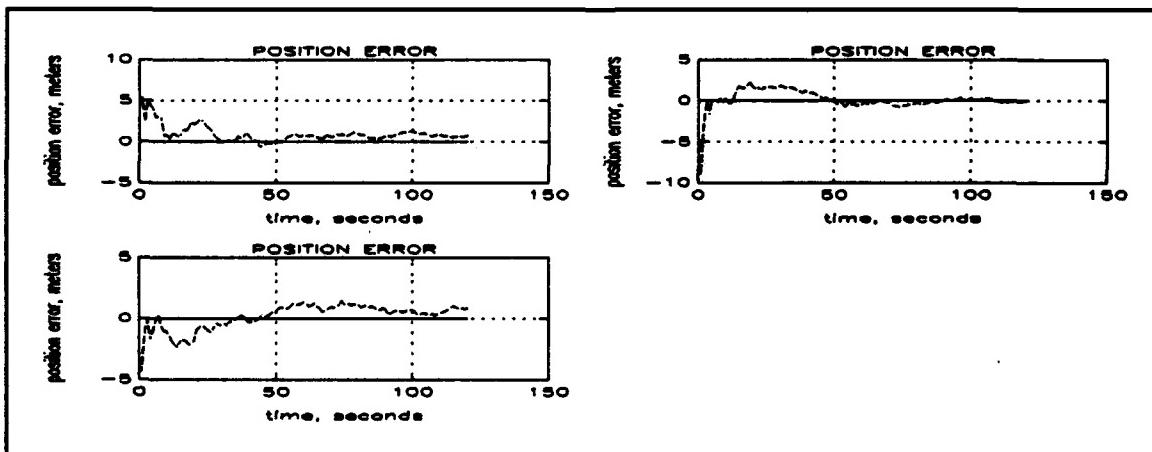


Figure 6.4: First-run, surface-interval position error.

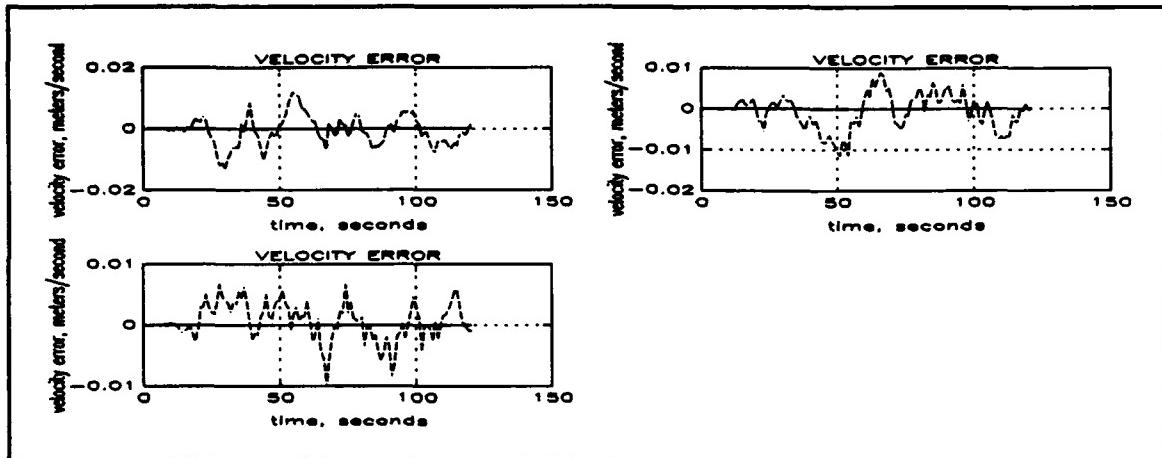


Figure 6.5: First-run, surface-interval velocity error.

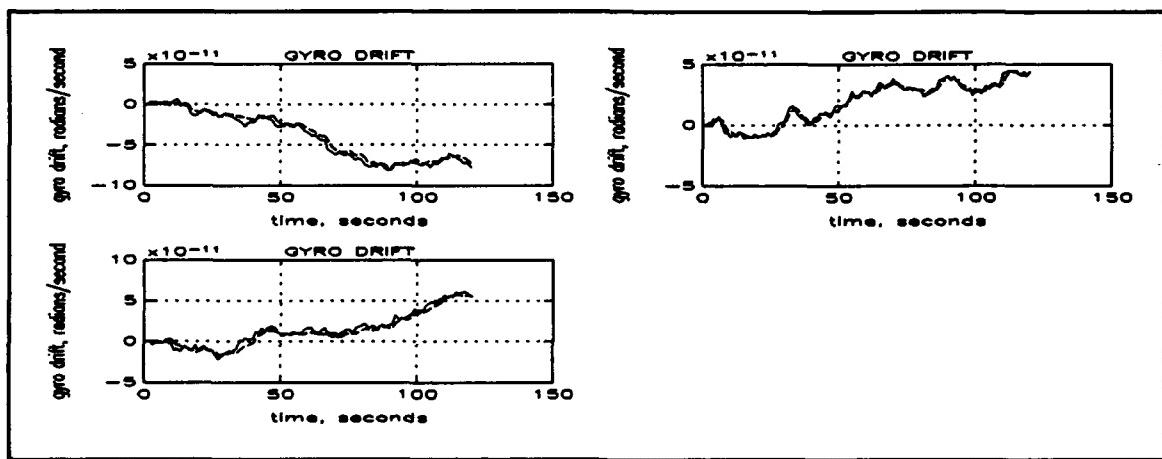


Figure 6.6: First-run, surface-interval gyro drift.

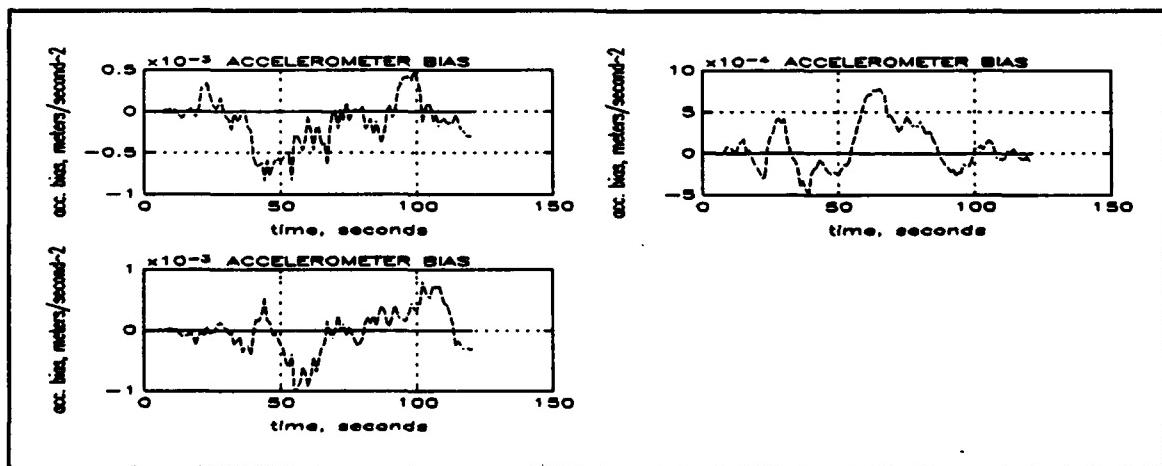


Figure 6.7: First-run, surface-interval accelerometer bias.

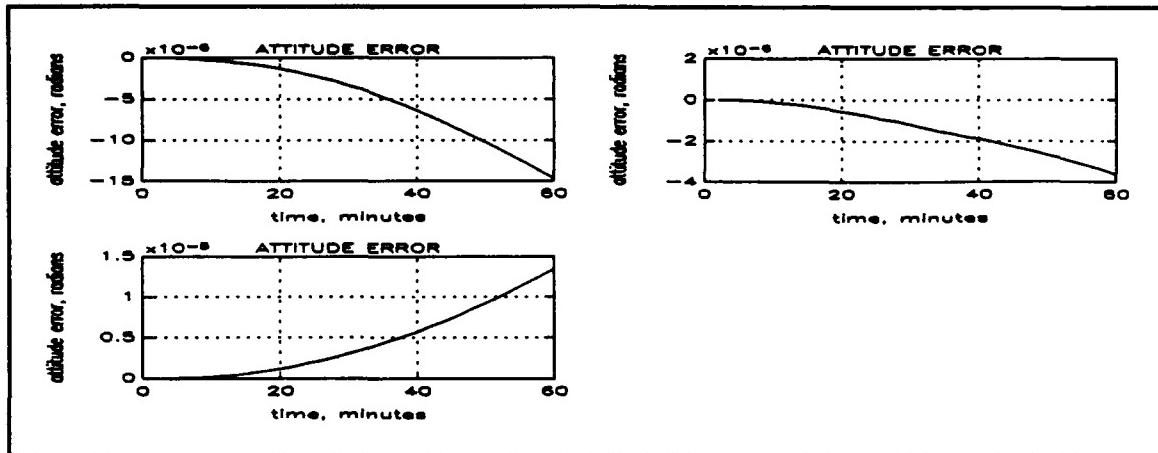


Figure 6.8: First-run, dive-series attitude error.

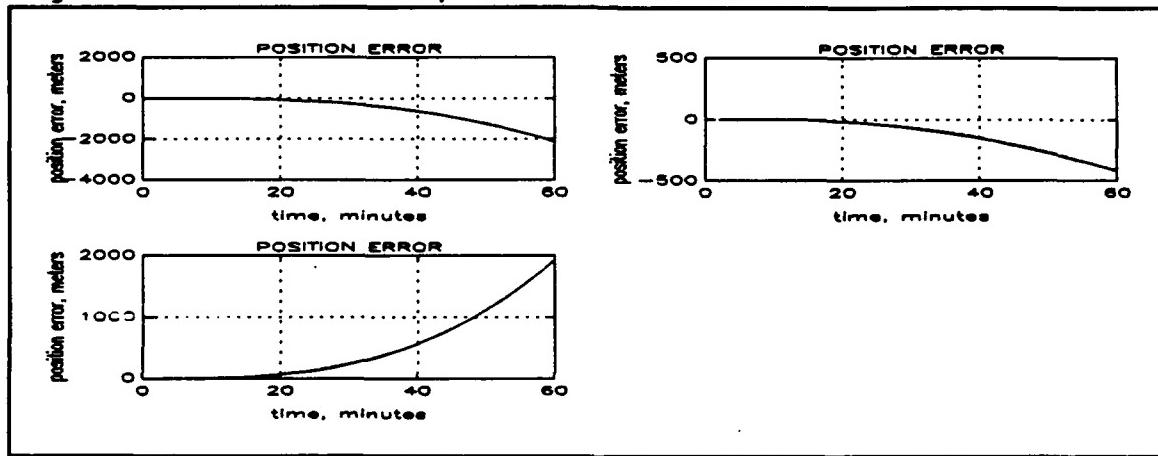


Figure 6.9: First-run, dive-series position error.

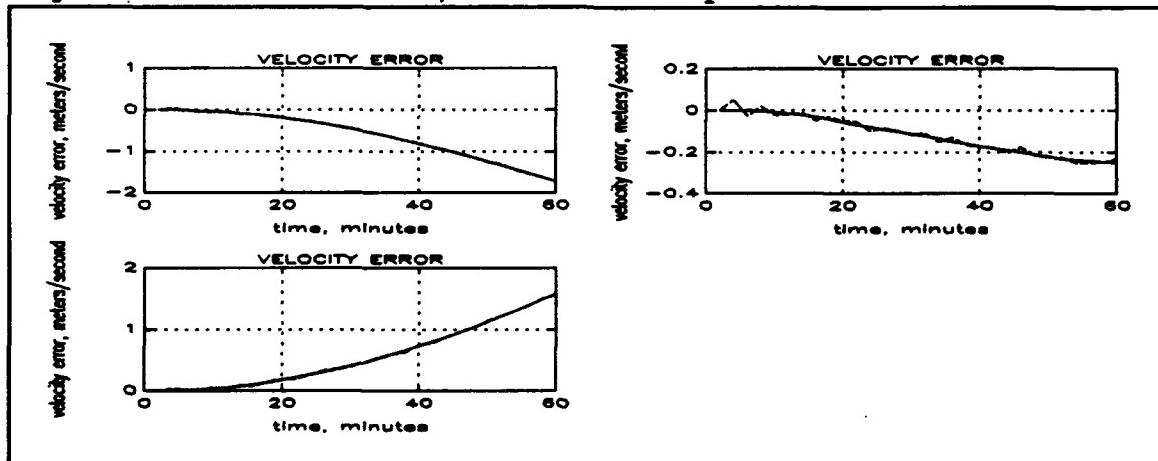


Figure 6.10: First-run, dive-series velocity error.

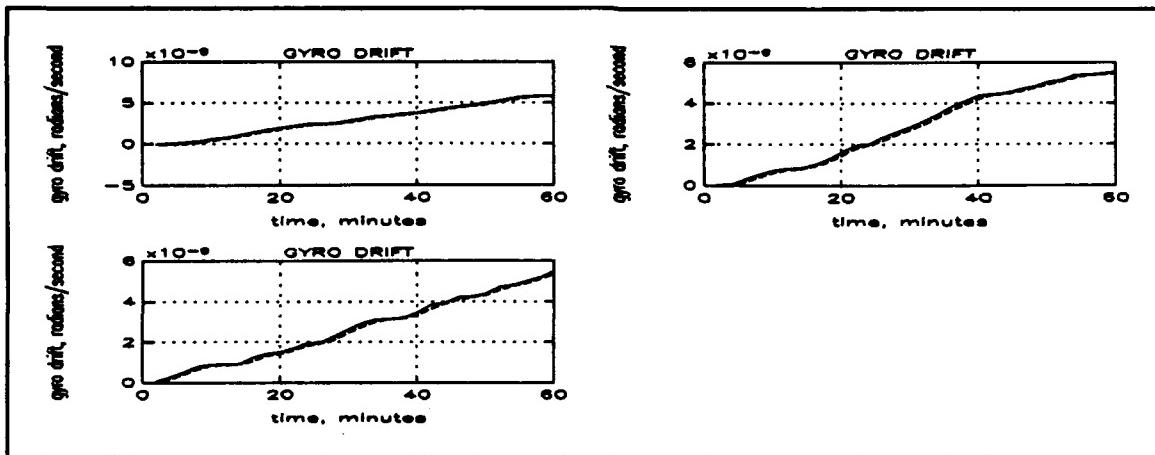


Figure 6.11: First-run, dive-series gyro drift.

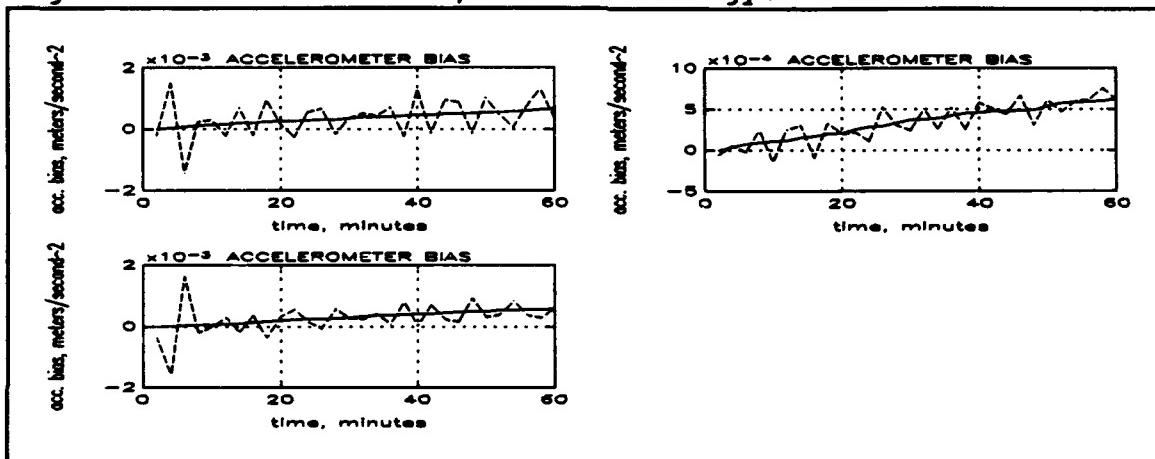


Figure 6.12: First-run, dive-series accelerometer bias.

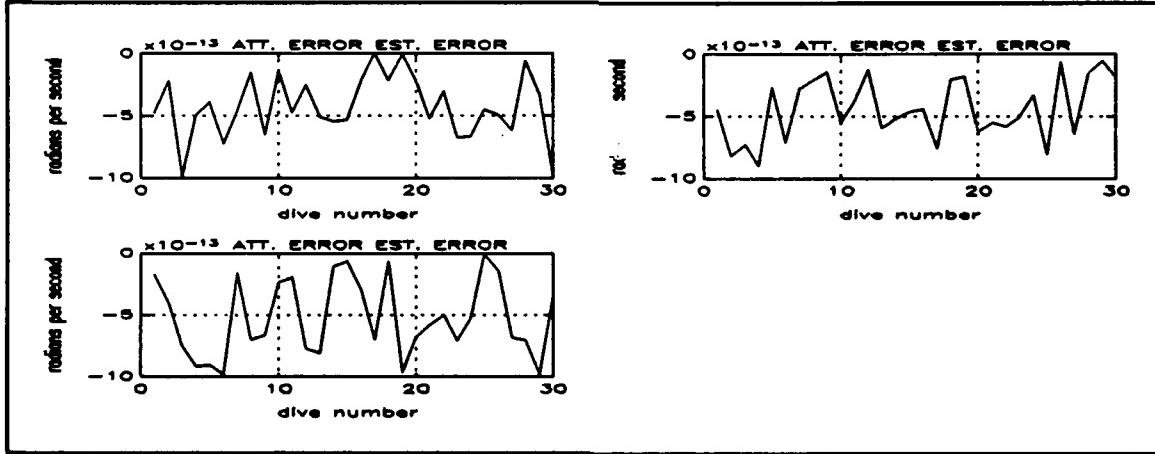


Figure 6.13: First-run attitude error estimate error.

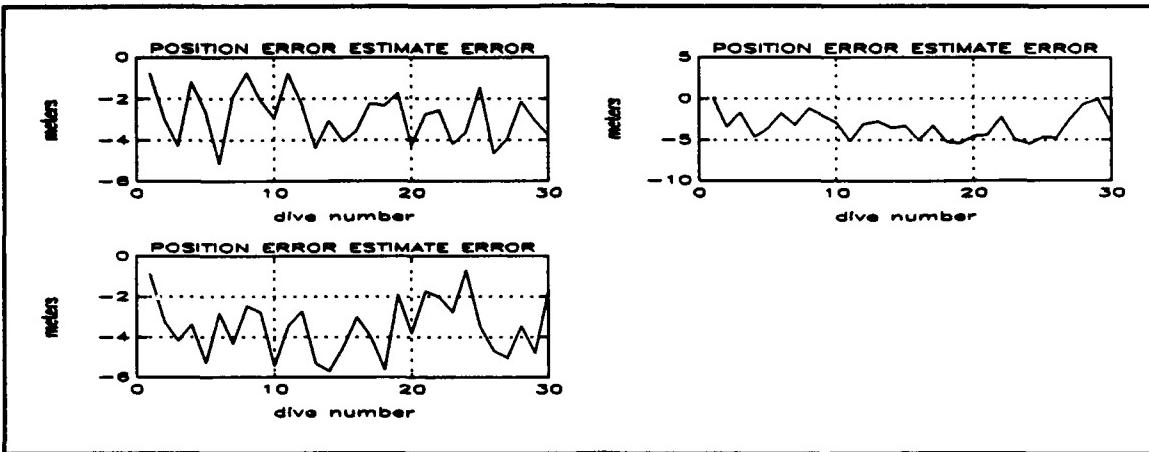


Figure 6.14: First-run position error estimate error.

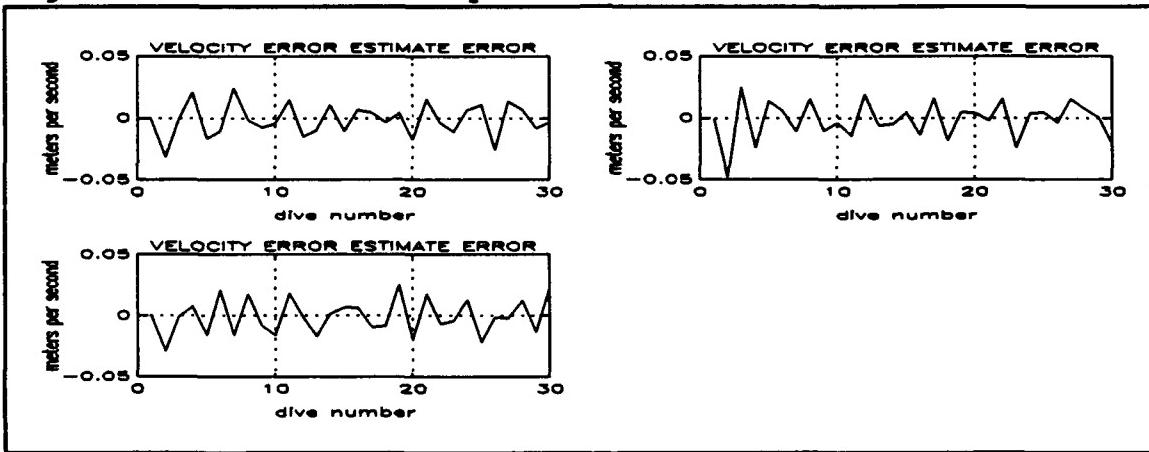


Figure 6.15: First-run velocity error estimate error.

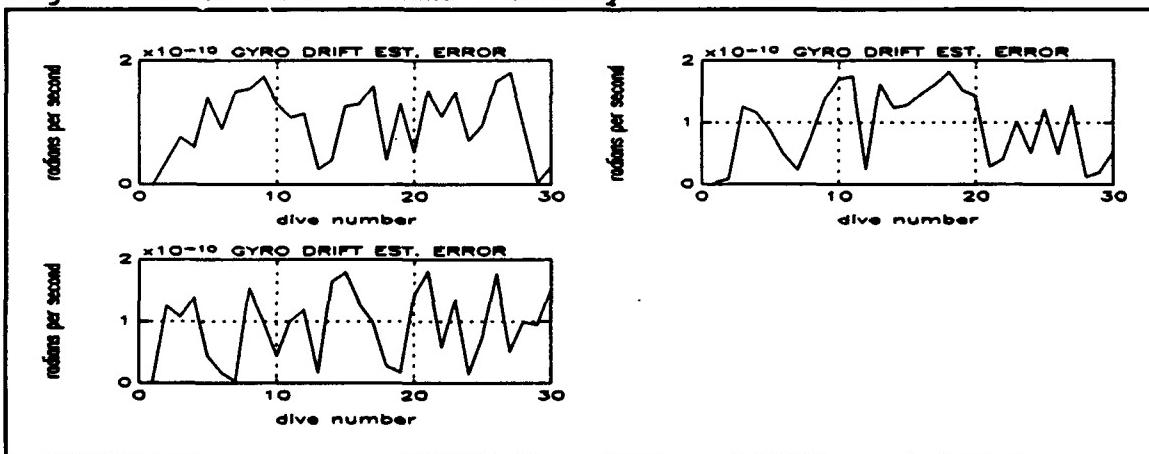


Figure 6.16: First-run gyro drift estimate error.

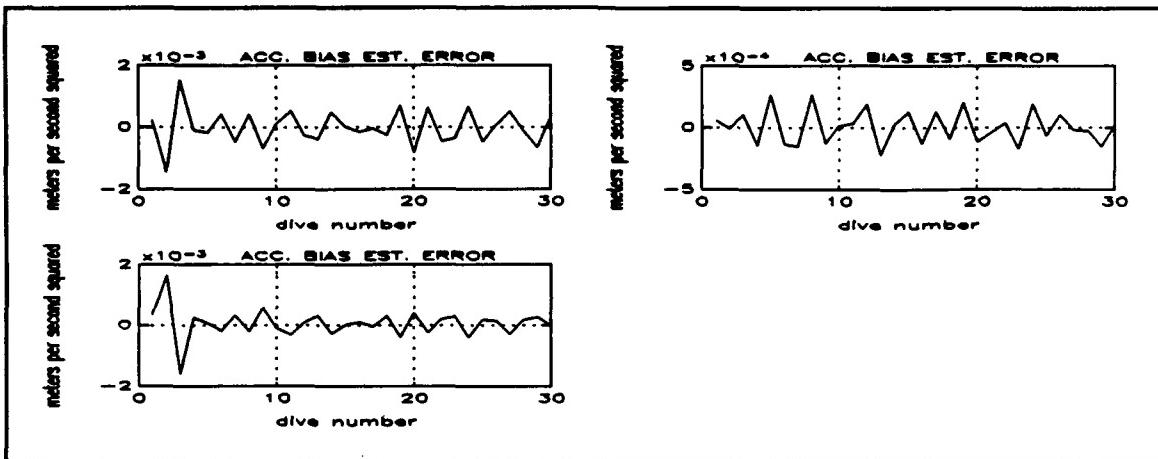


Figure 6.17: First-run accelerometer bias estimate error.

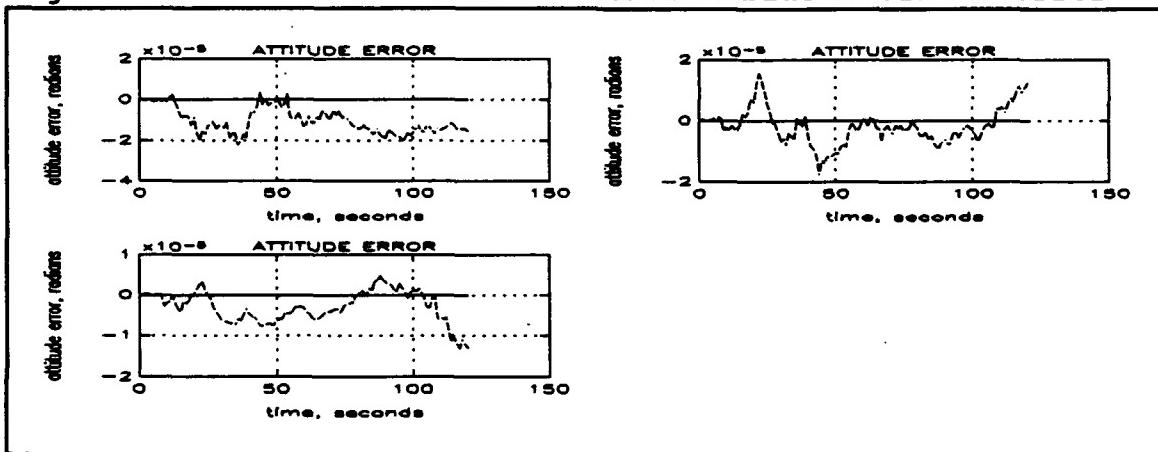


Figure 6.18: Second-run, surface-interval attitude error.

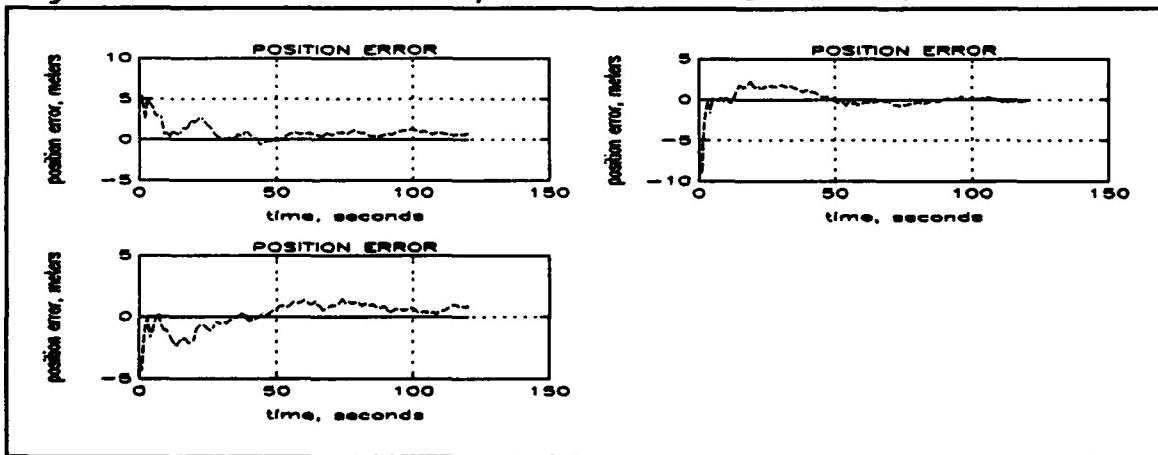


Figure 6.19: Second-run, surface-interval position error.

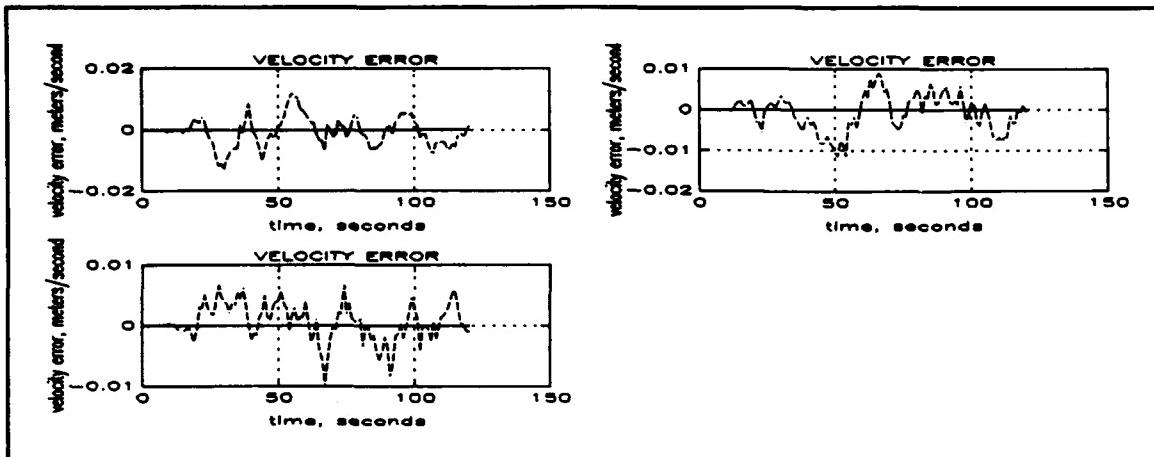


Figure 6.20: Second-run, surface-interval velocity error.

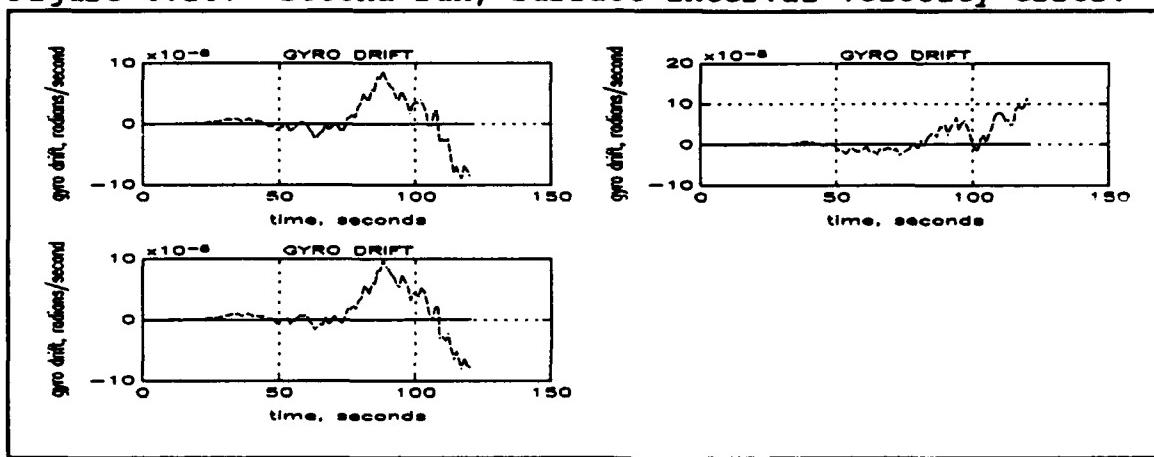


Figure 6.21: Second-run, surface-interval gyro drift.

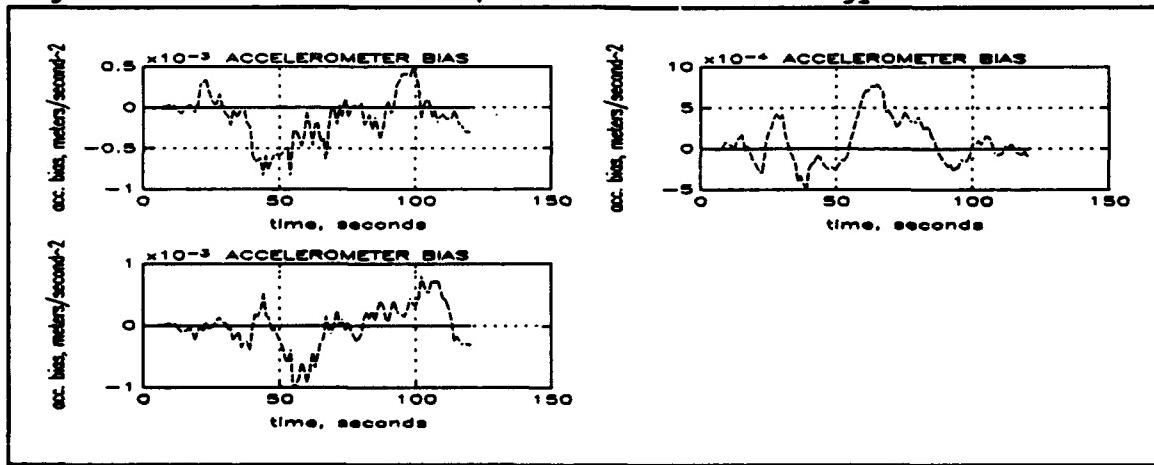


Figure 6.22: Second-run, surface-interval accelerometer bias.

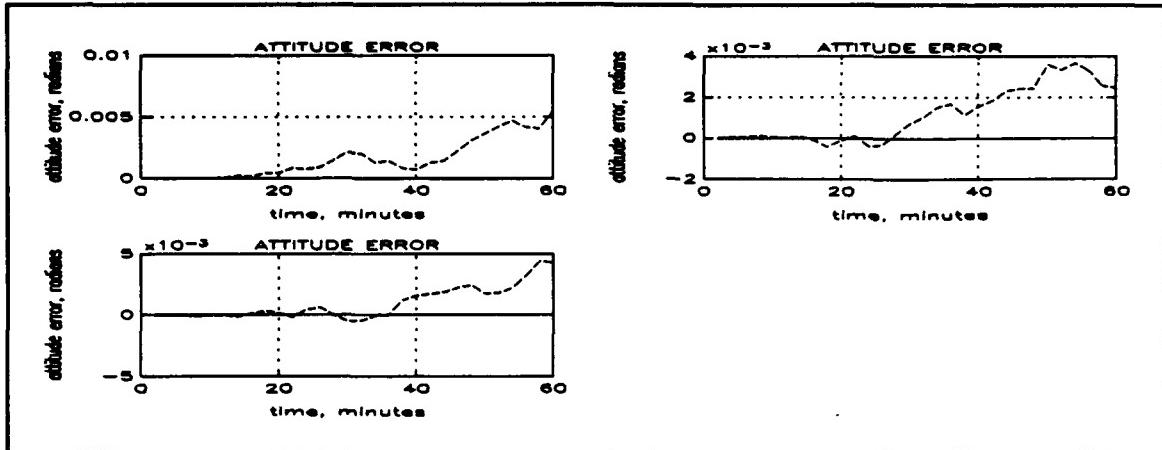


Figure 6.23: Second-run, dive-series attitude error.

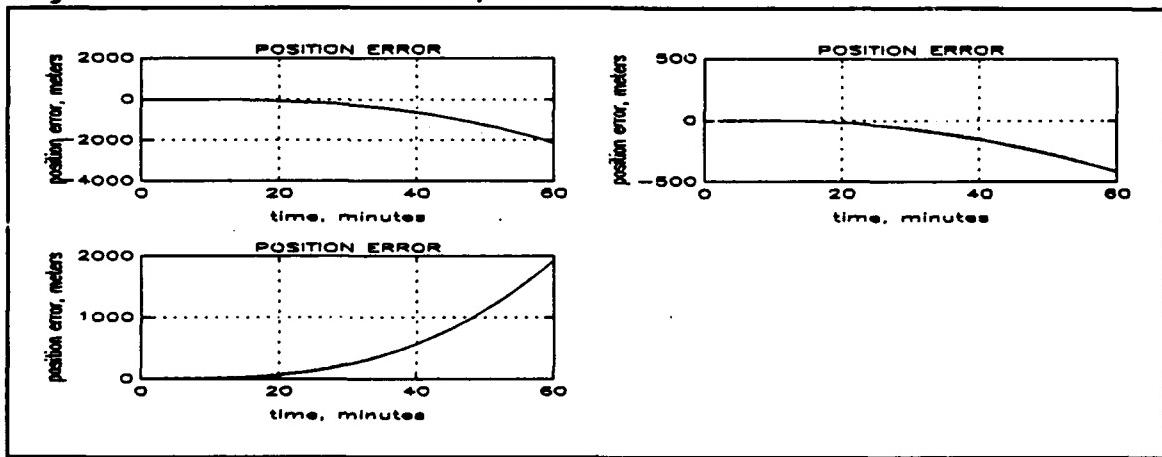


Figure 6.24: Second-run, dive-series position error.

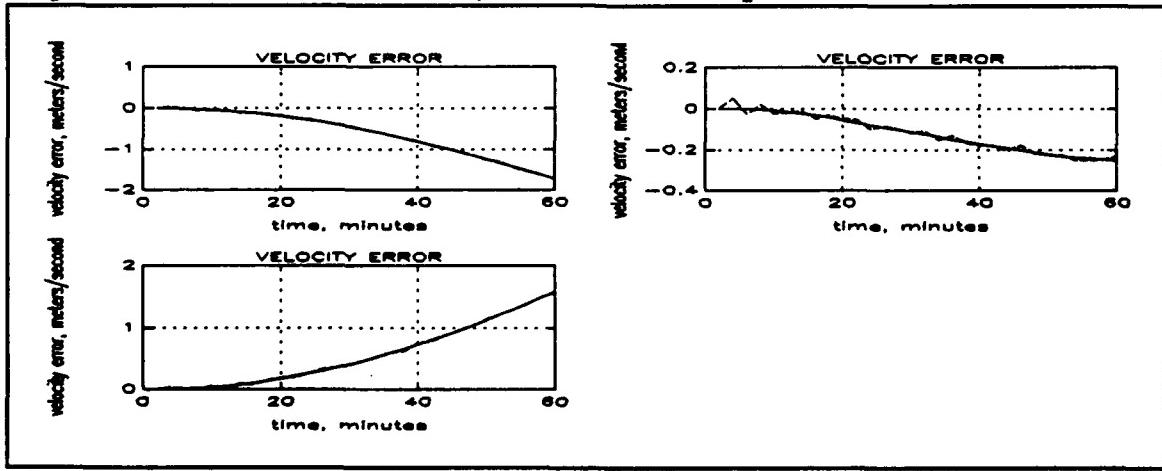


Figure 6.25: Second-run, dive-series velocity error.

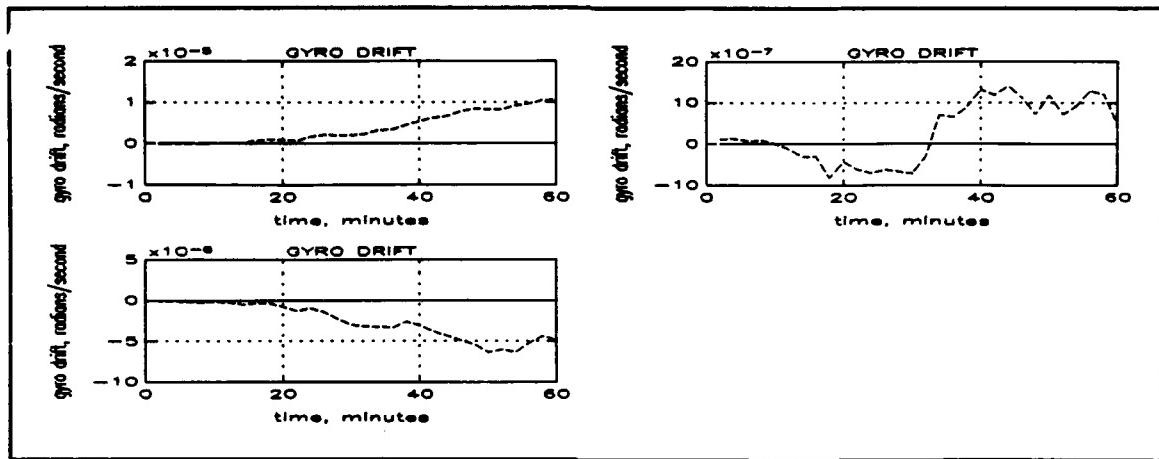


Figure 6.26: Second-run, dive-series gyro drift.

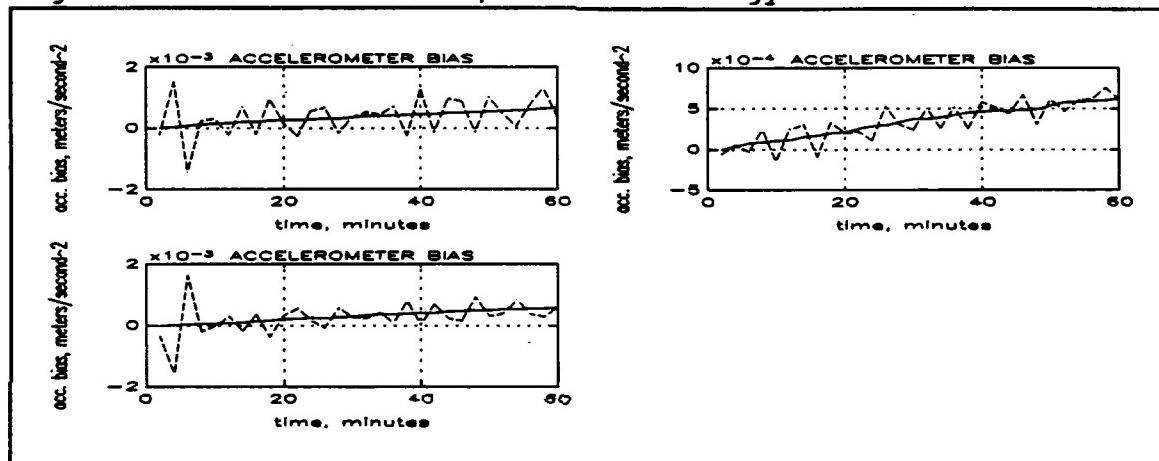


Figure 6.27: Second-run, dive-series accelerometer bias.

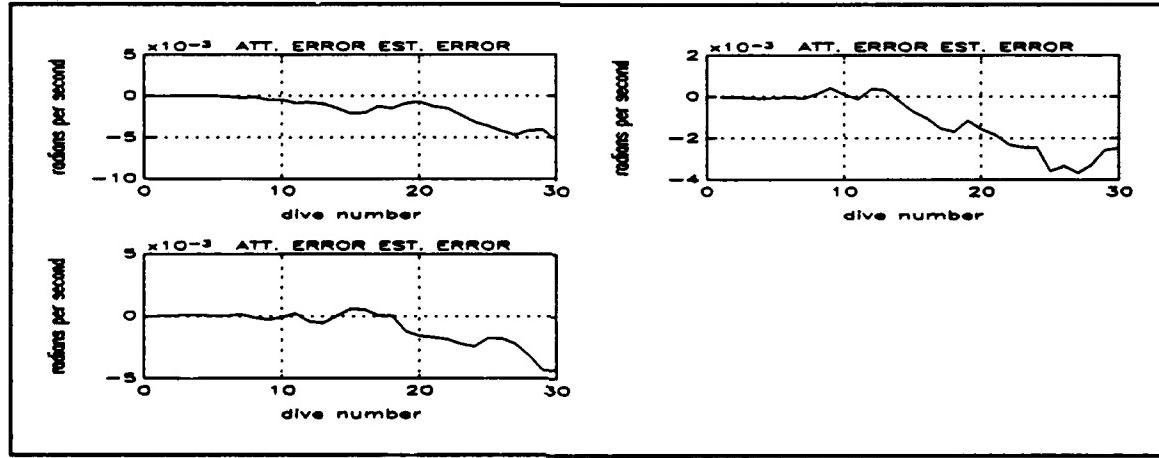


Figure 6.28: Second-run attitude error estimate error.

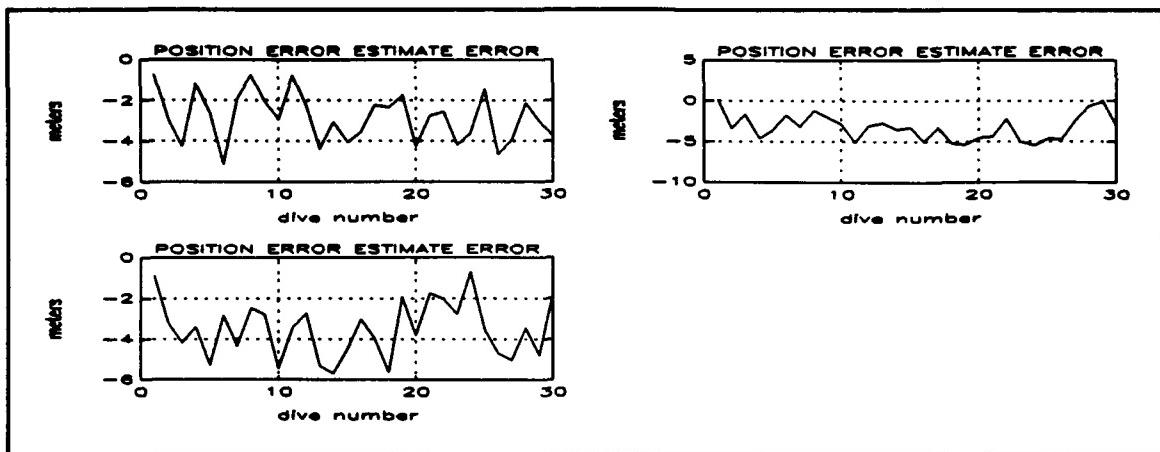


Figure 6.29: Second-run position error estimate error.

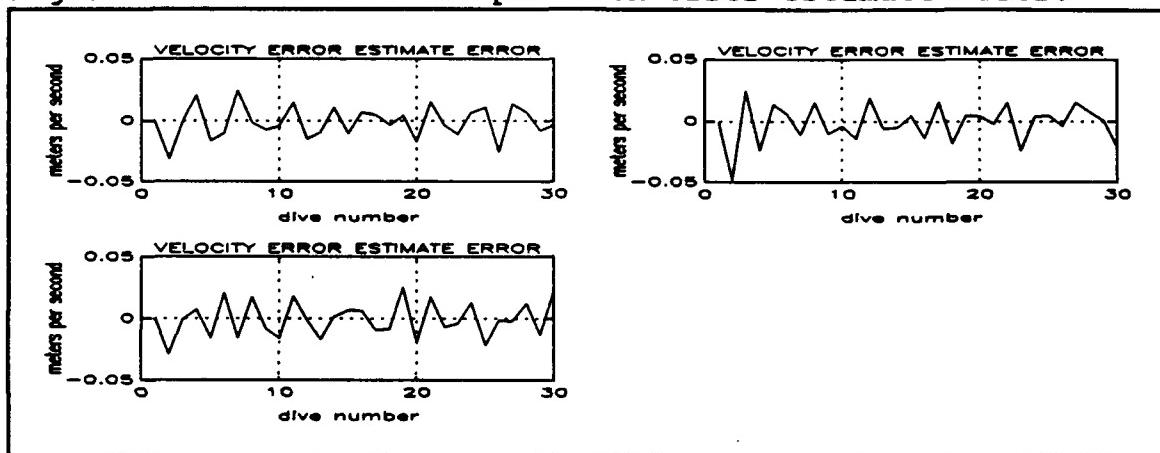


Figure 6.30: Second-run velocity error estimate error.

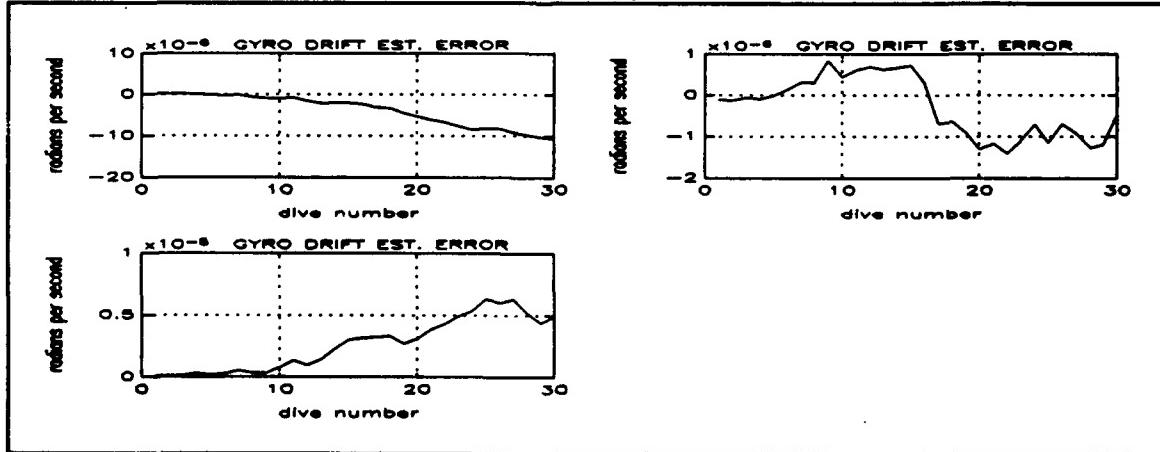


Figure 6.31: Second-run gyro drift estimate error.

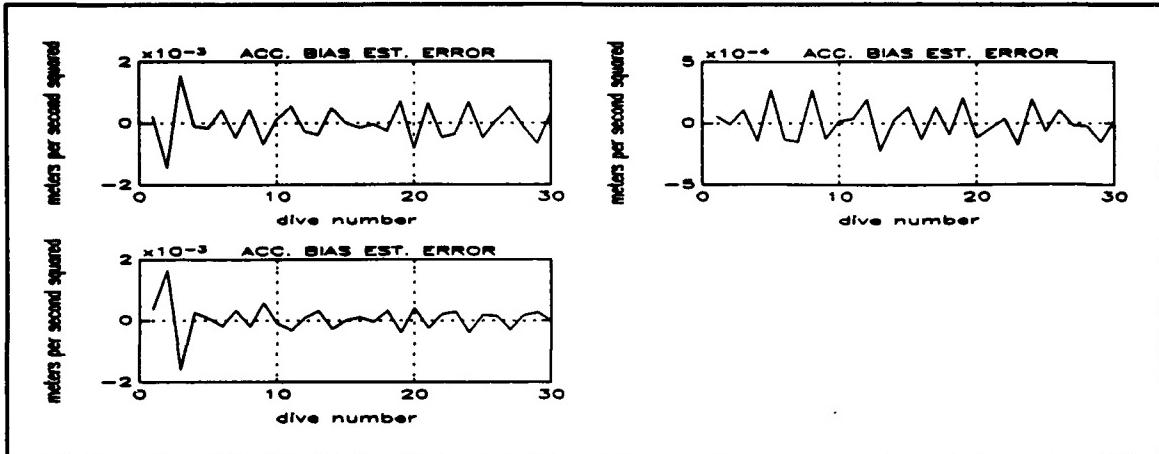


Figure 6.32: Second-run accelerometer bias estimate error.

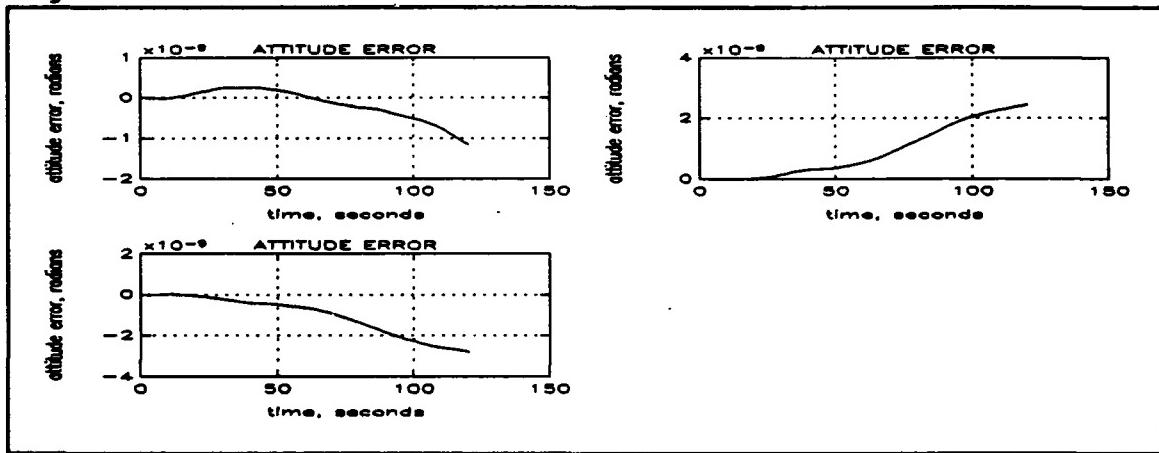


Figure 6.33: Third-run, surface-interval attitude error.

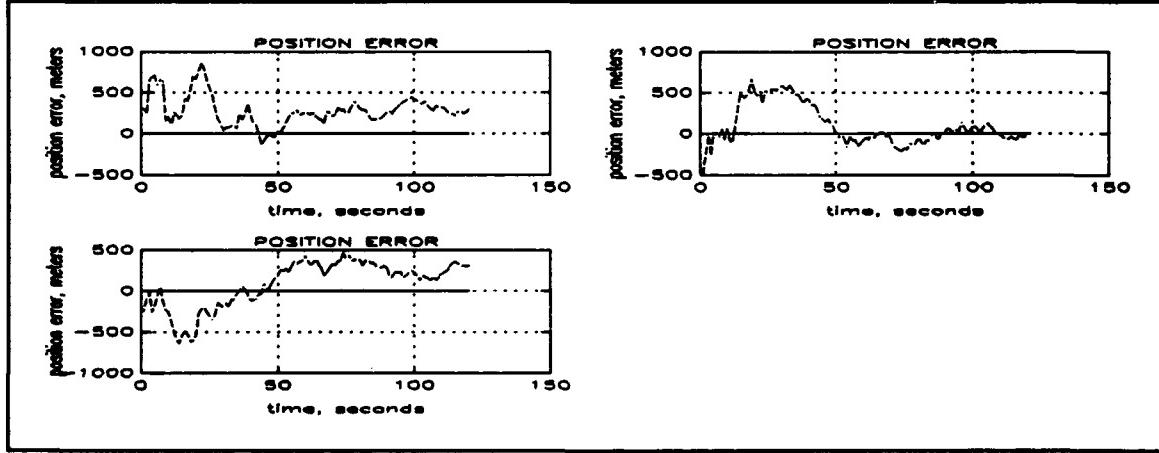


Figure 6.34: Third-run, surface-interval position error.

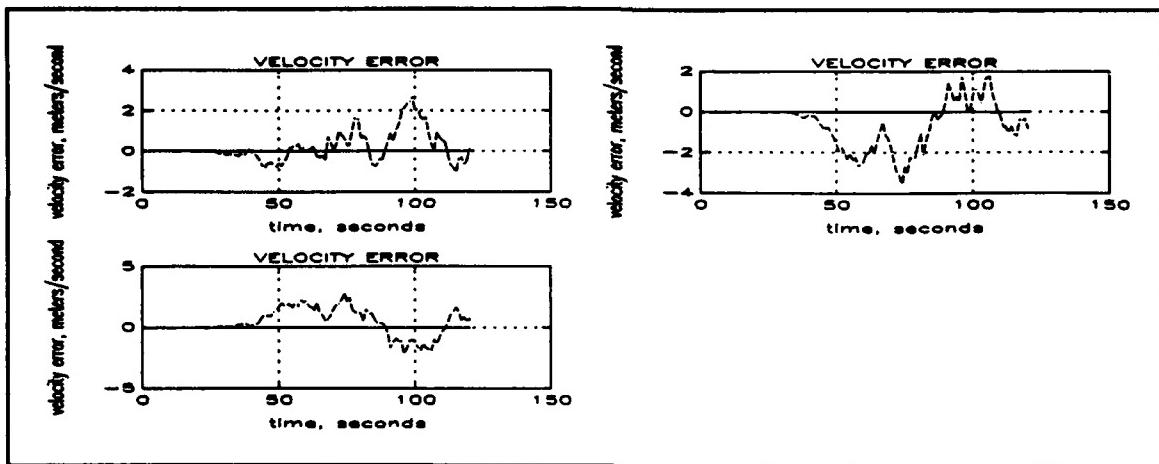


Figure 6.35: Third-run, surface-interval velocity error.

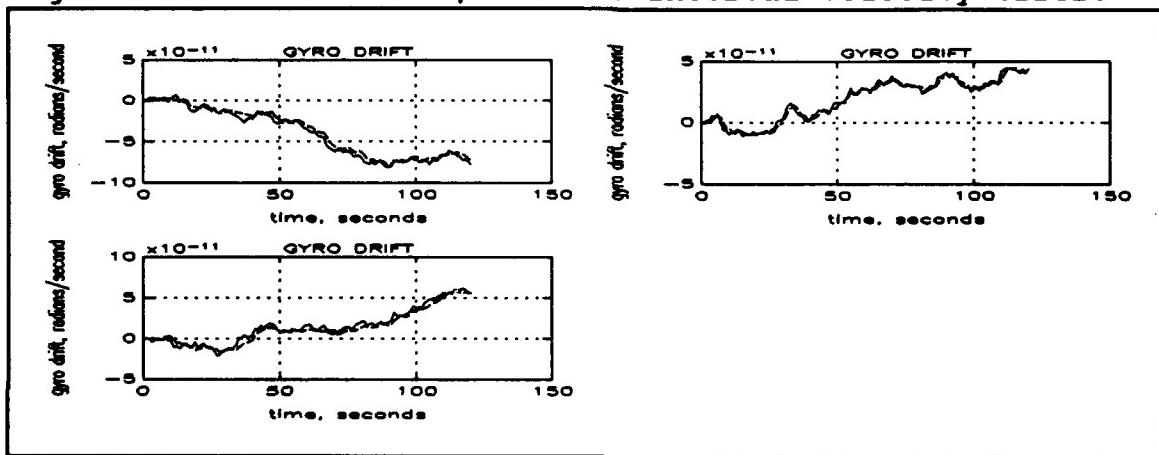


Figure 6.36: Third-run, surface-interval gyro drift.

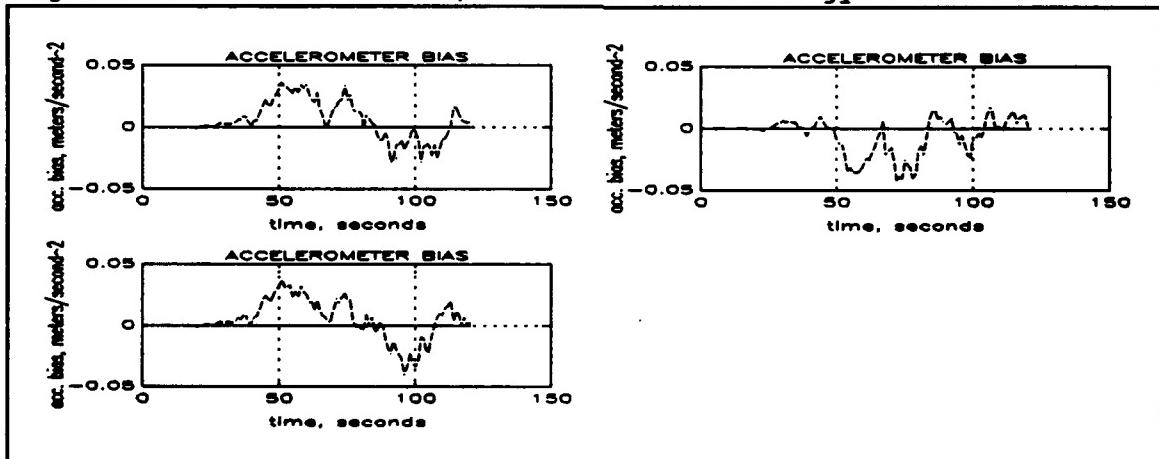


Figure 6.37: Third-run, surface-interval accelerometer bias.

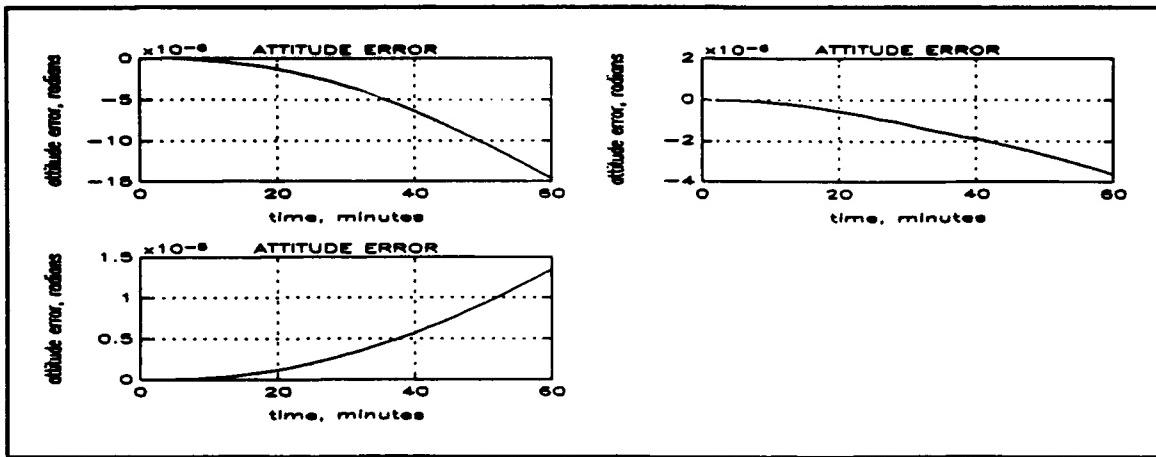


Figure 6.38: Third-run, dive-series attitude error.

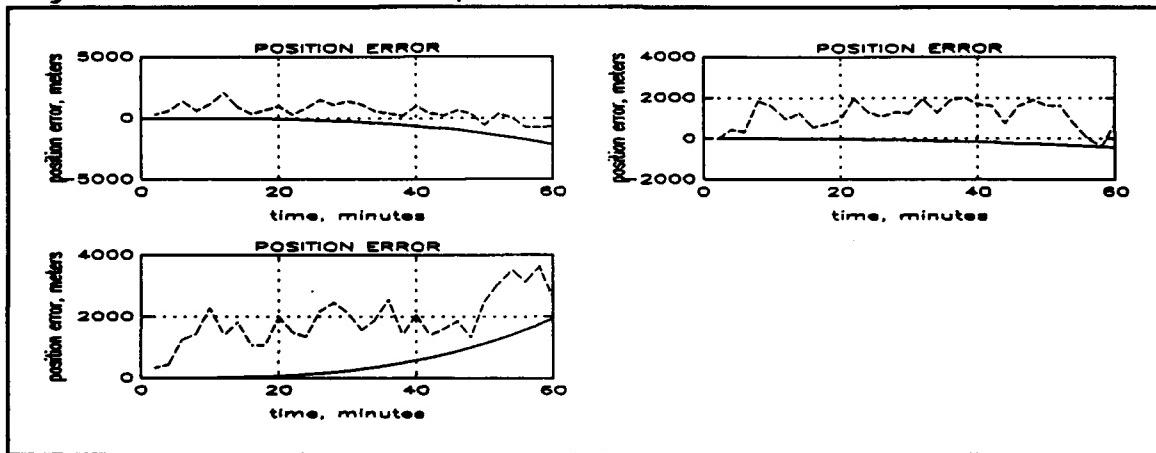


Figure 6.39: Third-run, dive-series position error.

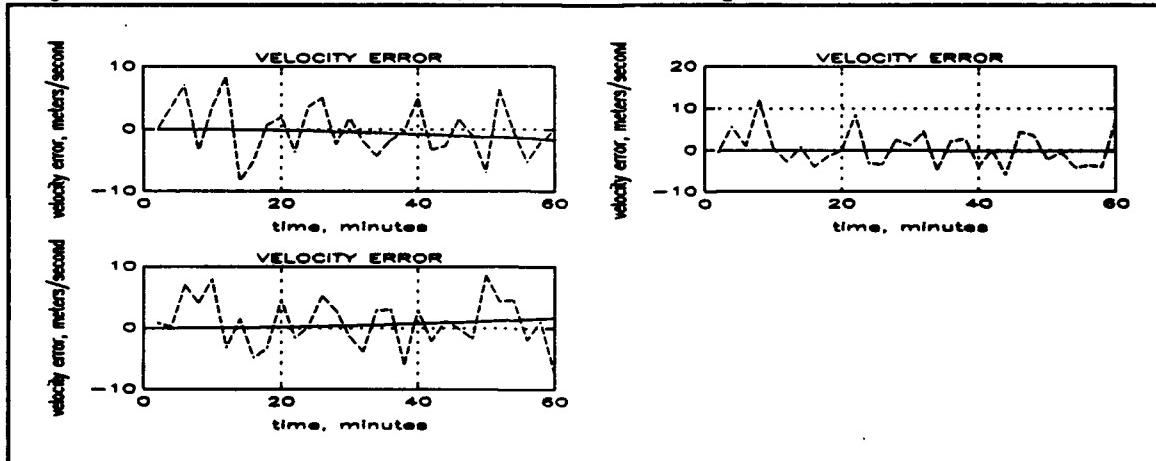


Figure 6.40: Third-run, dive-series velocity error.

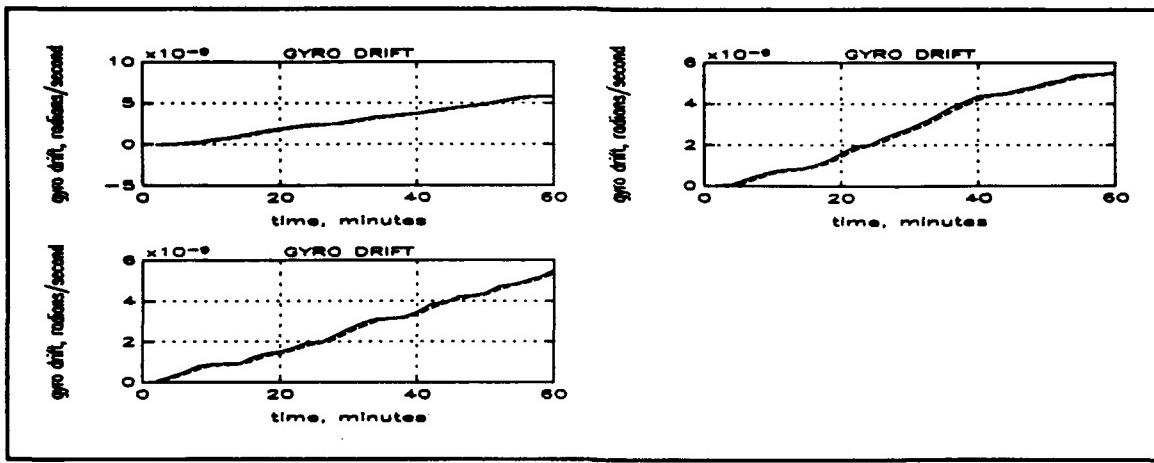


Figure 6.41: Third-run, dive-series gyro drift.

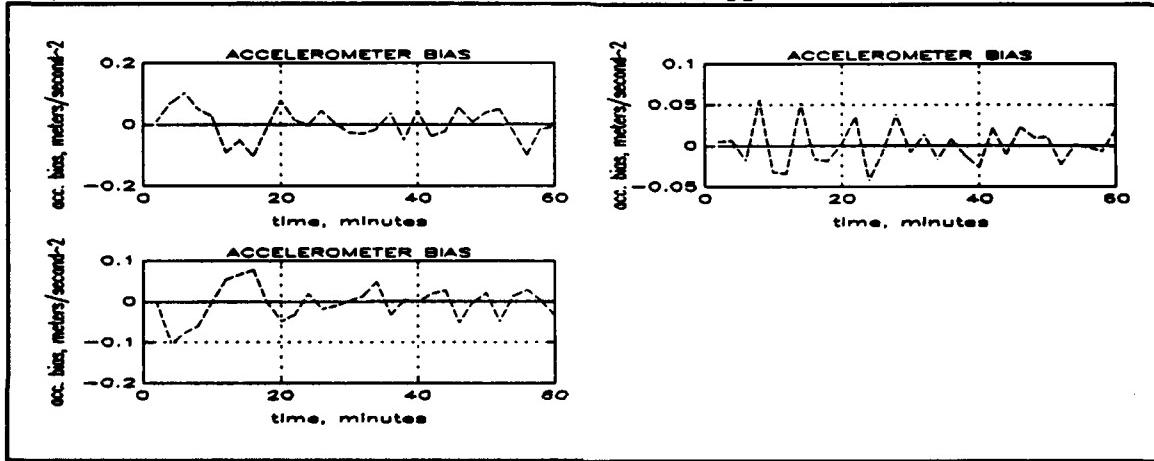


Figure 6.42: Third-run, dive-series accelerometer bias.

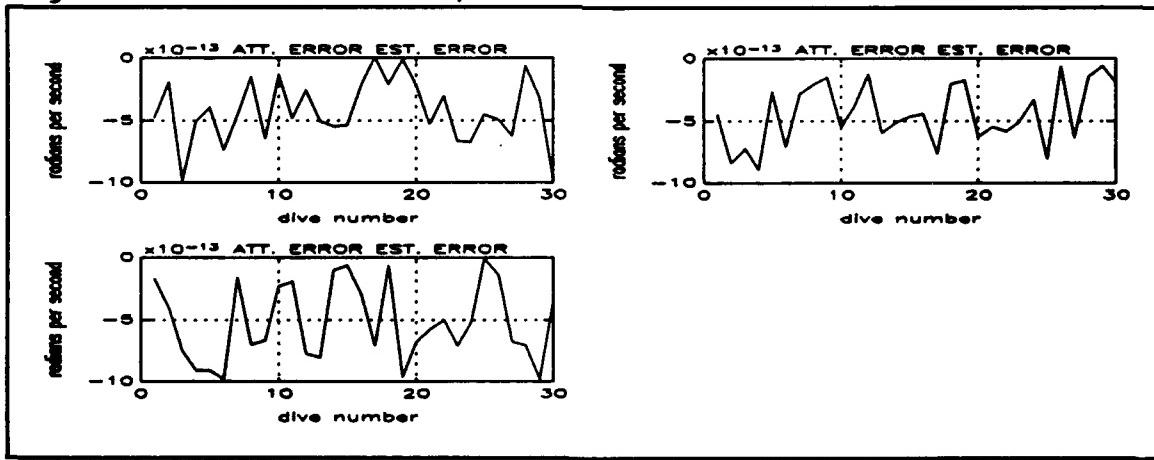


Figure 6.43: Third-run attitude error estimate error.

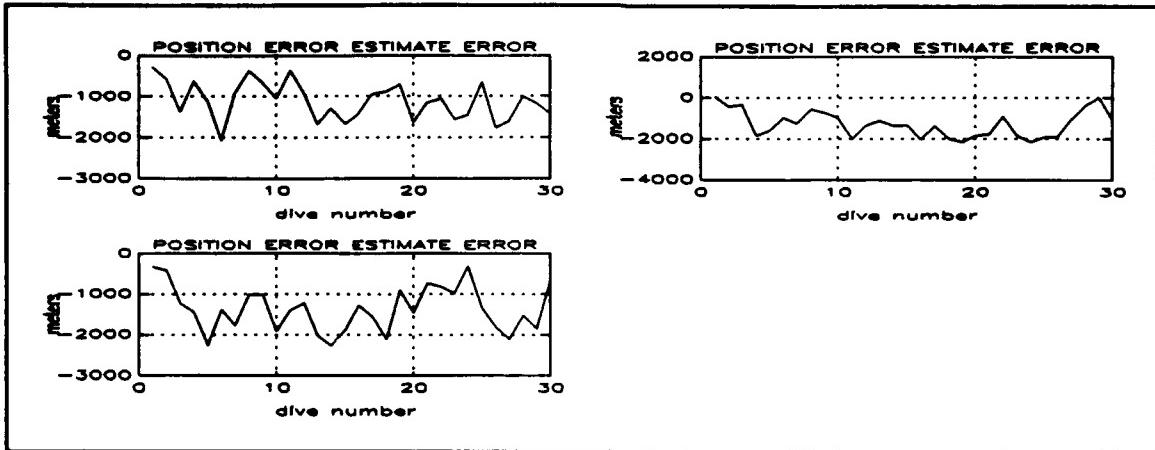


Figure 6.44: Third-run position error estimate error.

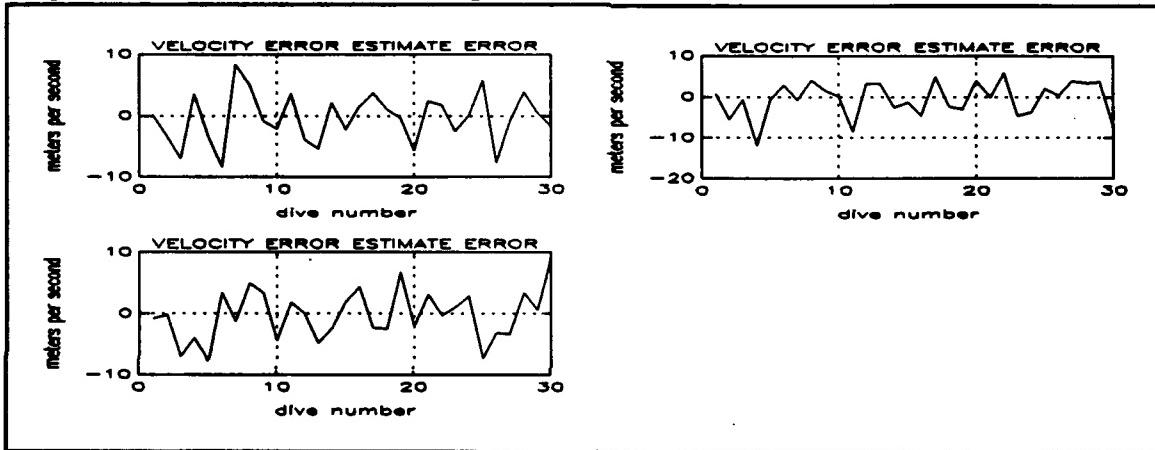


Figure 6.45: Third-run velocity error estimate error.

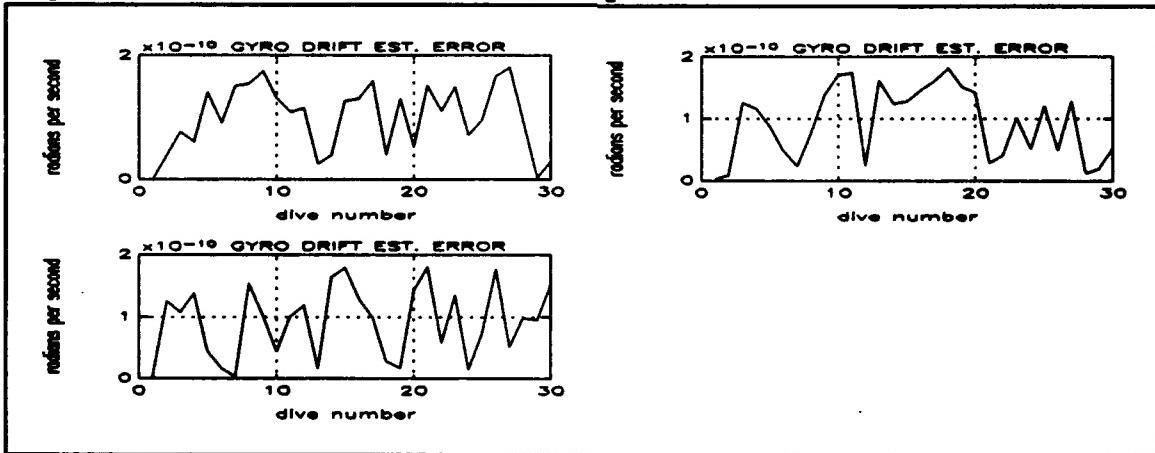


Figure 6.46: Third-run gyro drift estimate error.

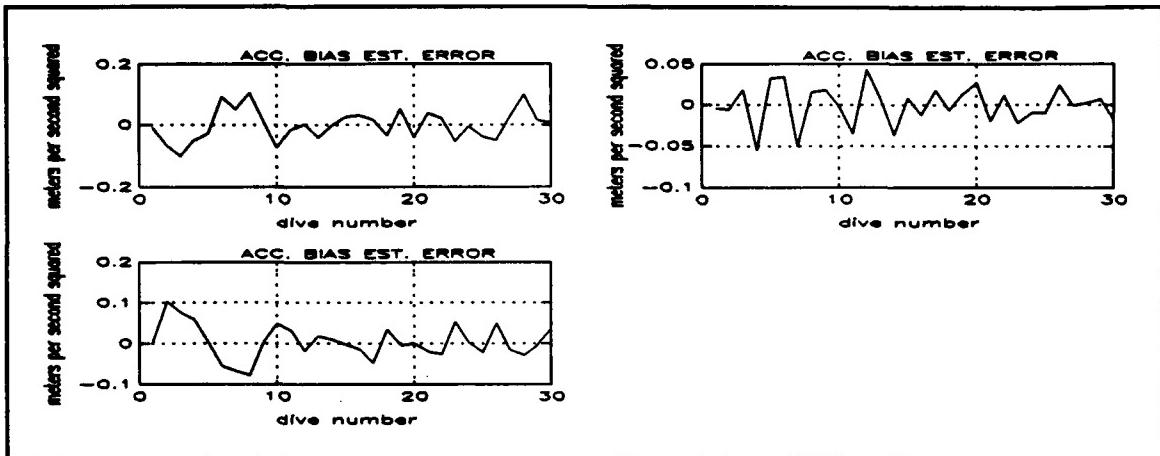


Figure 6.47: Third-run accelerometer bias estimate error.

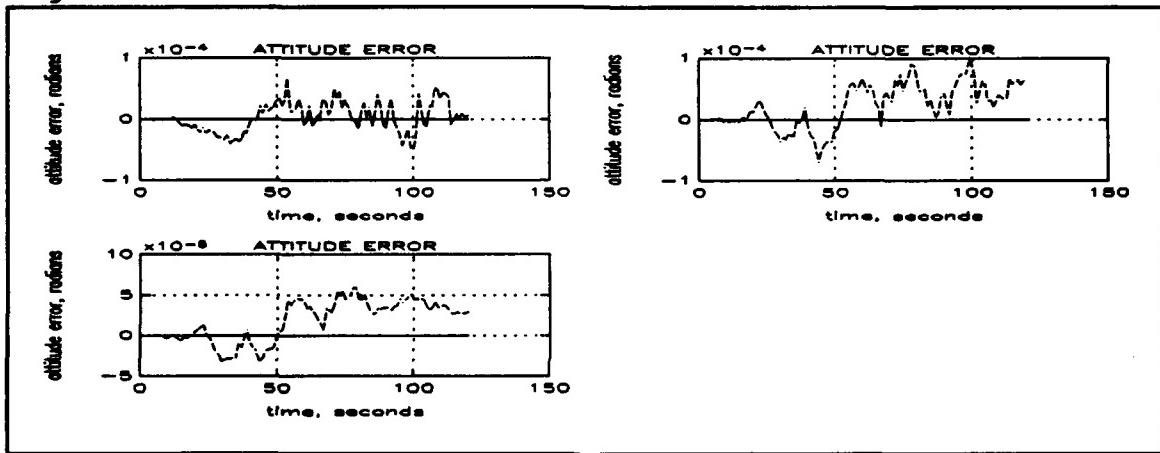


Figure 6.48: Fourth-run, surface-interval attitude error.

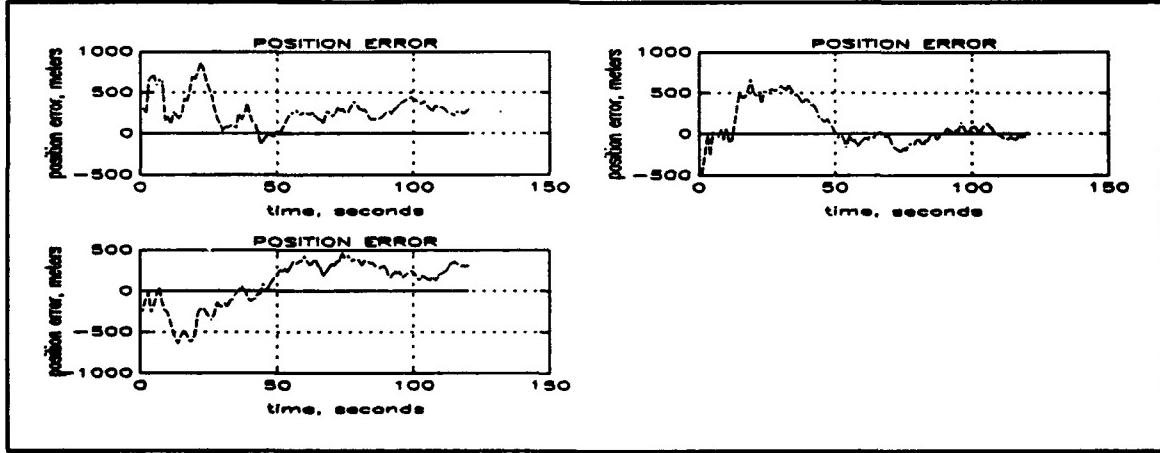


Figure 6.49: Fourth-run, surface-interval position error.

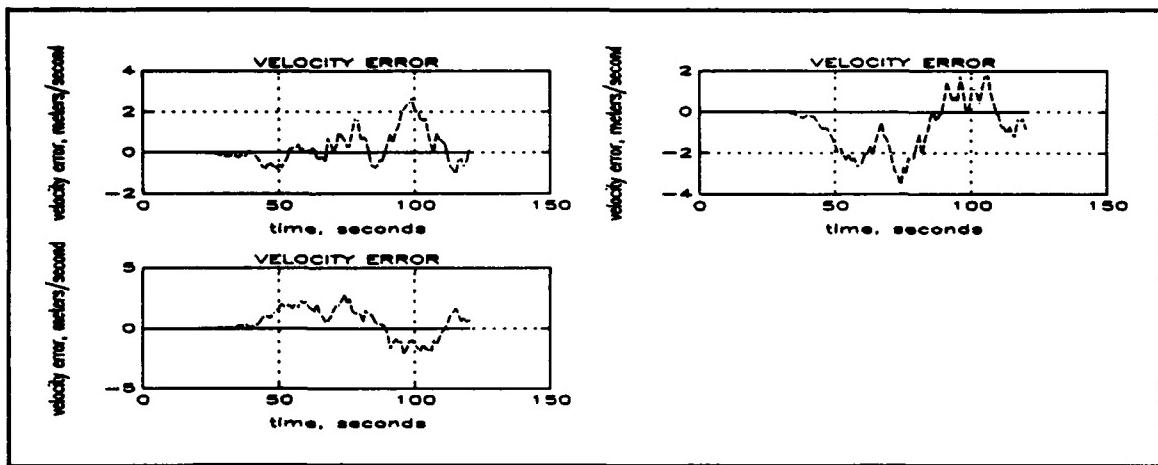


Figure 6.50: Fourth-run, surface-interval velocity error.

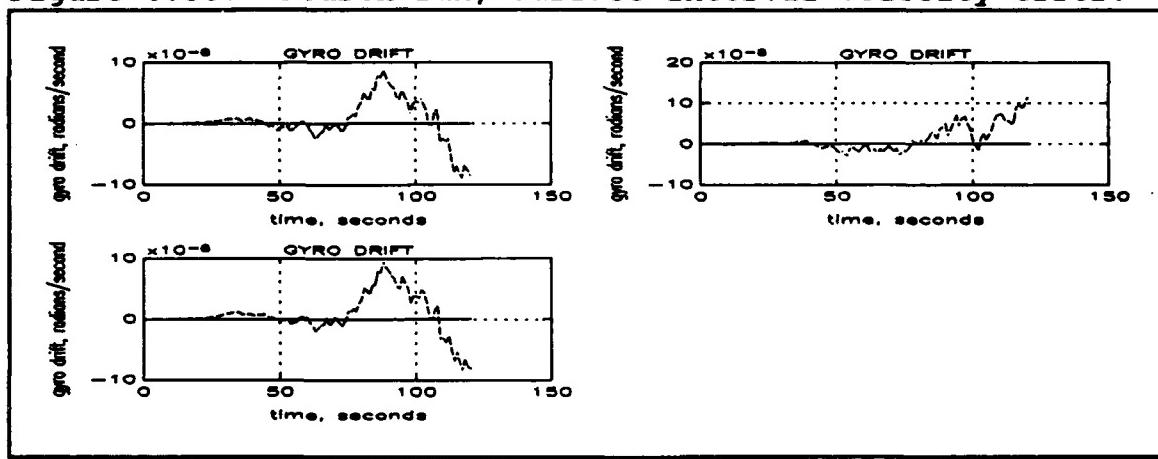


Figure 6.51: Fourth-run, surface-interval gyro drift.

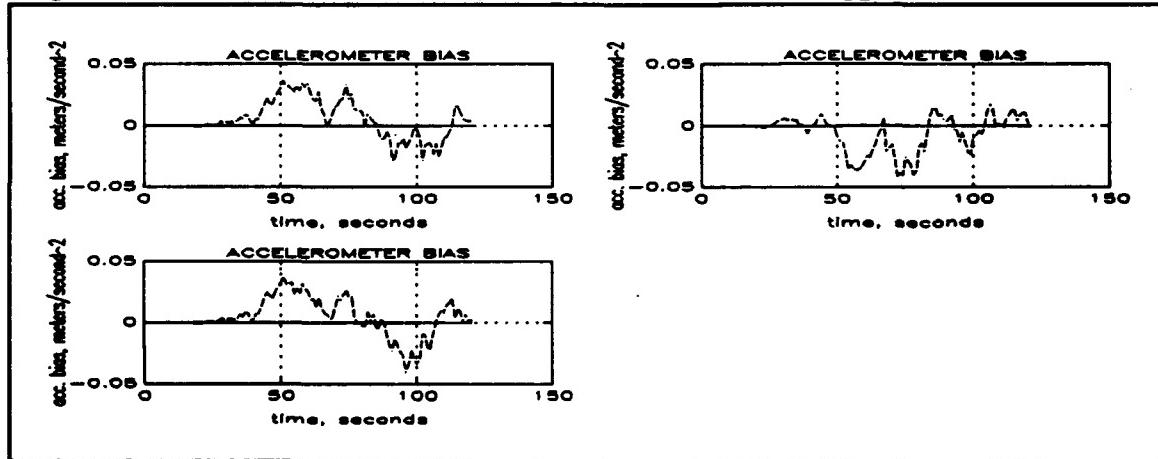


Figure 6.52: Fourth-run, surface-interval accelerometer bias.

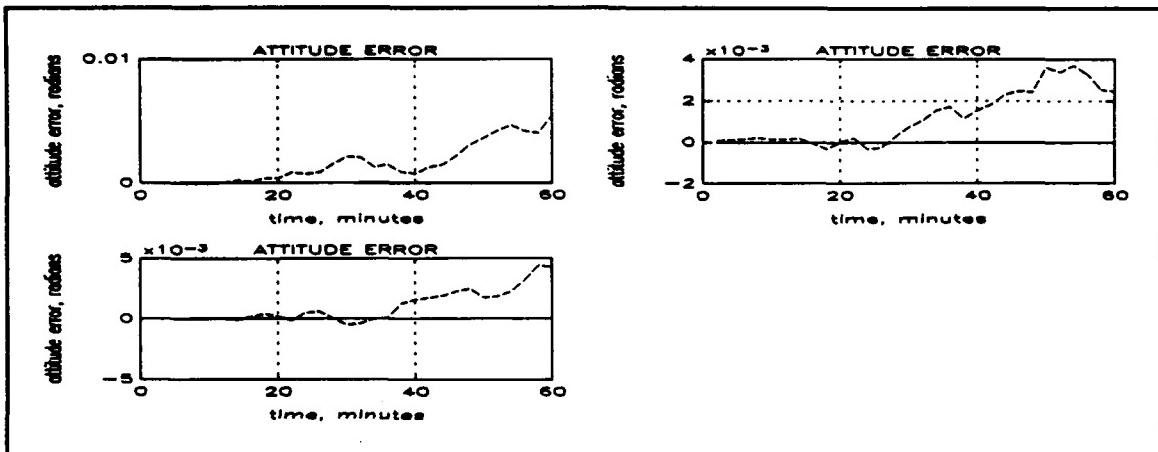


Figure 6.53: Fourth-run, dive-series attitude error.

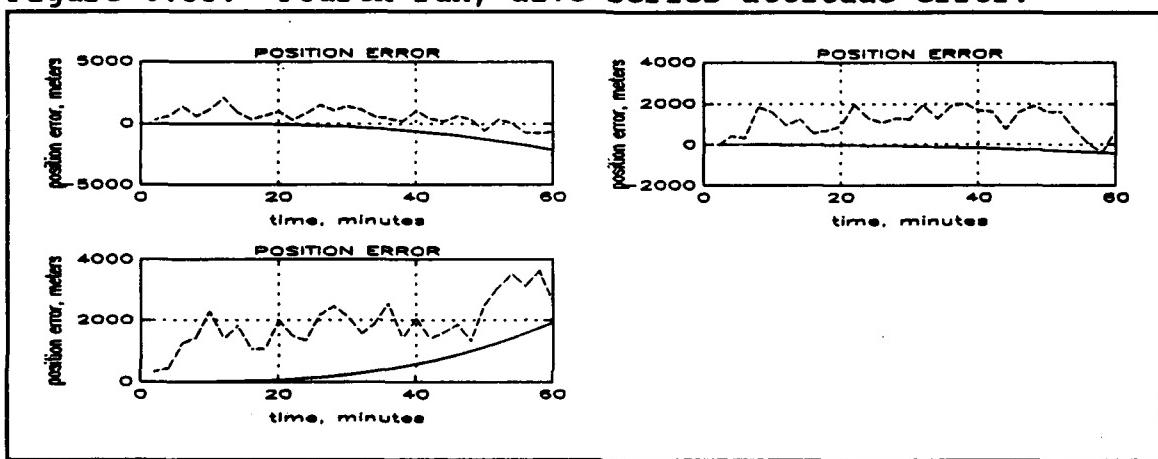


Figure 6.54: Fourth-run, dive-series position error.

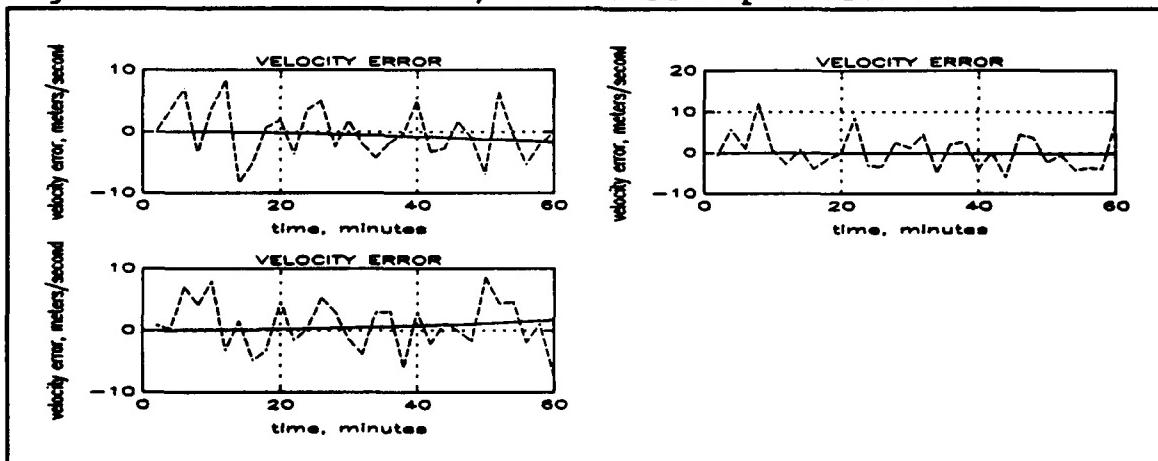


Figure 6.55: Fourth-run, dive-series velocity error.

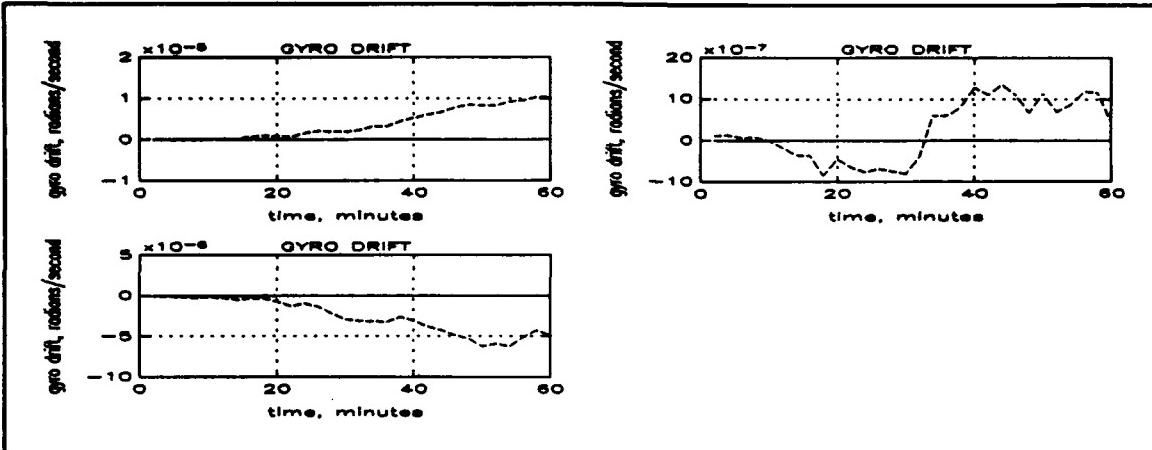


Figure 6.56: Fourth-run, dive-series gyro drift.

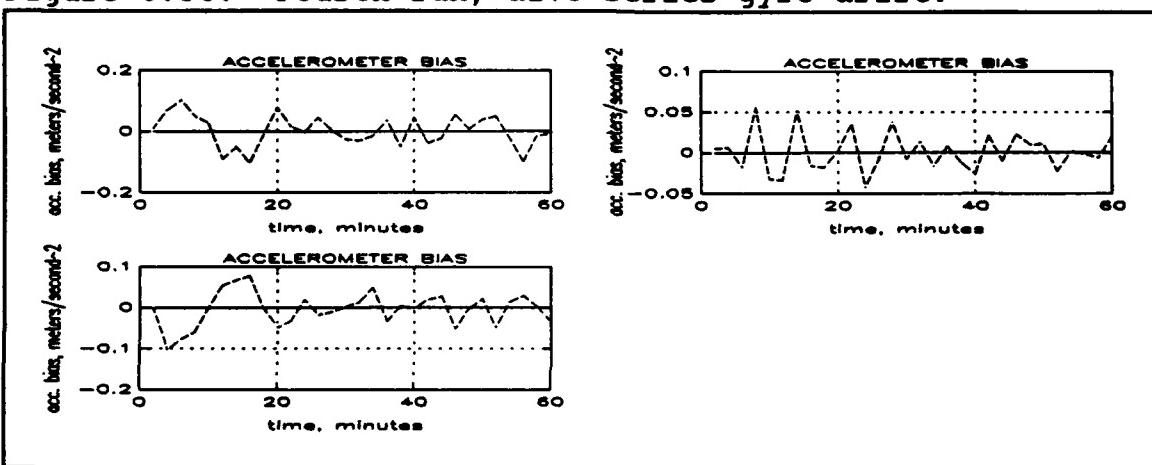


Figure 6.57: Fourth-run, dive-series accelerometer bias.

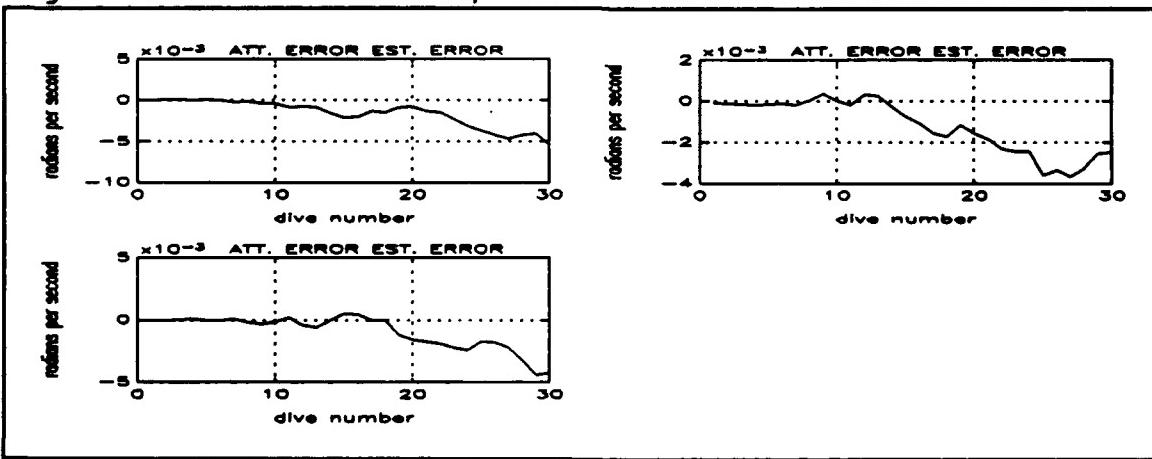


Figure 6.58: Fourth-run attitude error estimate error.

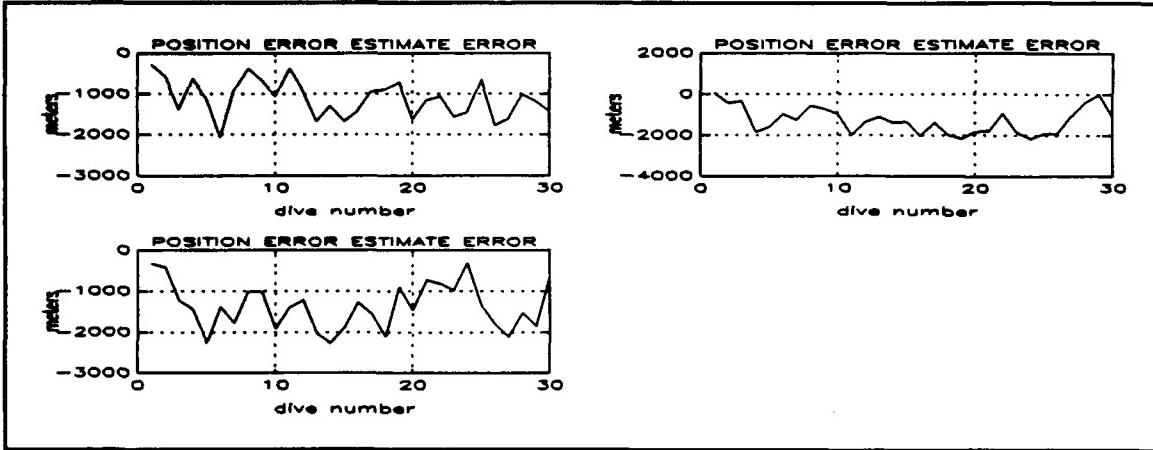


Figure 6.59: Fourth-run position error estimate error.

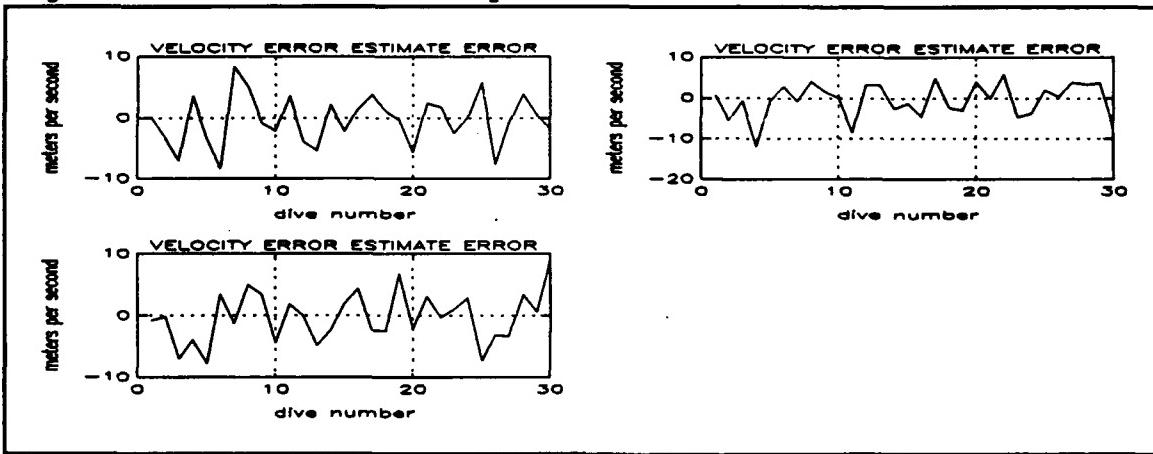


Figure 6.60: Fourth-run velocity error estimate error.

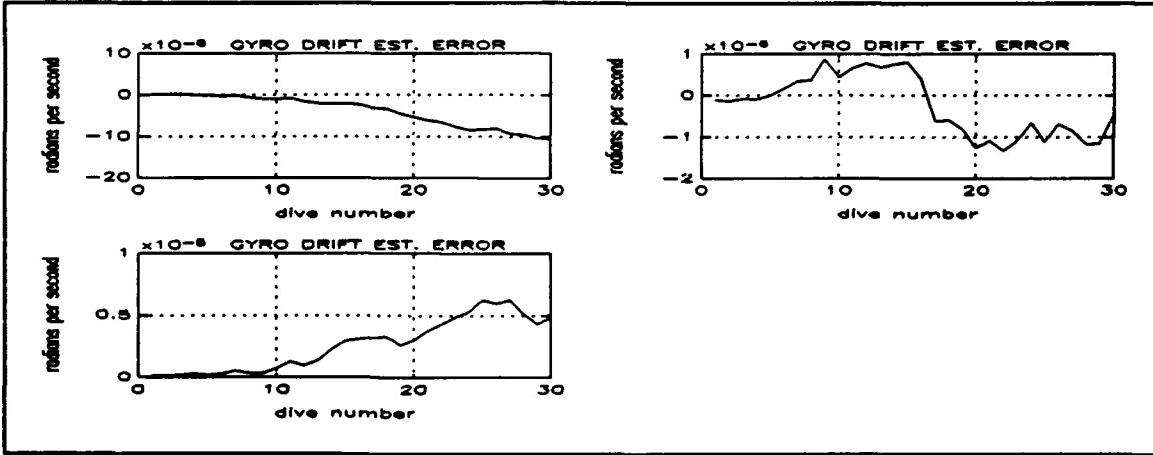


Figure 6.61: Fourth-run gyro drift estimate error.

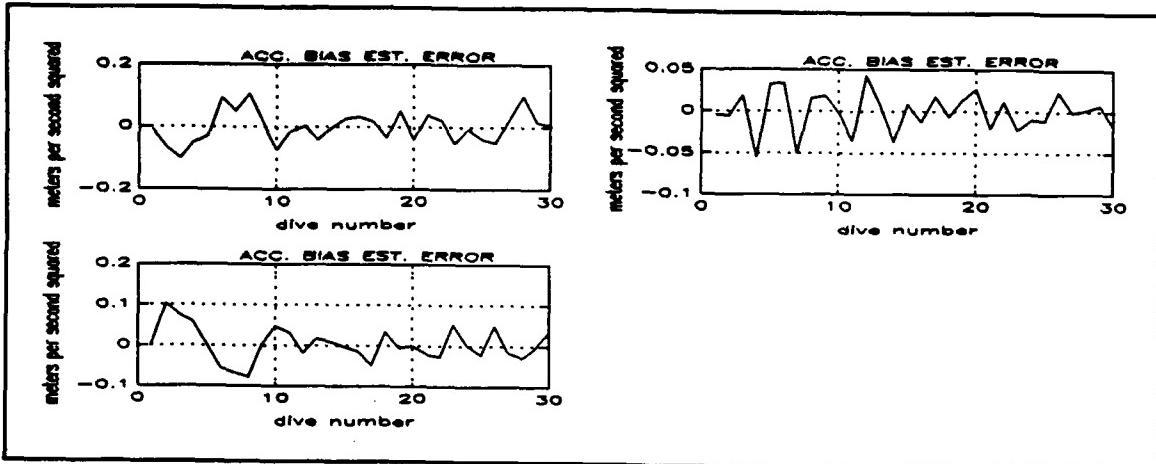


Figure 6.62: Fourth-run accelerometer bias estimate error.

Conclusions based on these results are included in the next chapter.

VII. CONCLUSIONS AND RECOMMENDATIONS

This research has produced a computer model of the integration of GPS and INS measurements for the navigation of an AUV. Noisy INS measurements were simulated with a model based on a medium-grade INS that uses RLG's. Noisy GPS and attitude measurements were simulated and subtracted from the modeled INS measurements to simulate INS errors. Kalman filtering was used to estimate those errors. Four simulations were run using combinations of high and low accuracy GPS and attitude measurements.

The results from the simulation runs matched the expectations. This can be observed by comparing Figure 6.9 with Figure 3.4. The simulated position error after one hour is on the order of one nautical mile, which is what is predicted by the theoretical equations.

The simulation shows that Kalman filtering can estimate the position and velocity errors to the expected accuracy levels. Figures 6.14, 6.15, 6.29, and 6.30 show that the accuracy of the position and velocity error estimates is independent of the accuracy of the attitude measurements.

The computer model can be used to compare combinations of INS's and GPS equipment with different levels of accuracy. These comparisons might include low-grade or high-grade INS's and P-code GPS receivers.

The recommended approach to continuing this research would begin with a verification of the INS error model. This could be accomplished using the AUV in the pool at the Naval Postgraduate School. A trajectory measurement using the AUV's sonars could be used as a standard to determine typical INS errors. Simulations would need to be run with a profile compatible with the dimensions of the pool.

Additional research could investigate implementation with closed-loop utilization of state estimates and smoothing for improved accuracy.

APPENDIX A. GPS DATA PROCESSING SOFTWARE

unpack.for:

```
c      unpack mx4200 data
c
c      character*11 Rtype
c      character*14 fname, outfname
c      write (*,'(a\')') ' What is the input file name? '
c      read (*,'(a')') fname
c      open (1, file=fname, access='sequential',
c +status='old')
c      write (*,'(a\')') ' What is the output file name? '
c      read (*,'(a')') outfname
c      open (2, file=outfname, status='new')
c      startTIME = 0
c      incH = 0
c      incLAT = 0
c      incLONG = 0
100   continue
c      read (1,'(a)',end=500,err=100) Rtype
c      write (*,'(a')') Rtype
c      if (Rtype .eq. '$PMVXG,001,') then
c          backspace 1
c          read (1,50,err=100) Rtype,ih,im,is,ideg,dmin,
c +ideglong,dminlong,alt,isrc
50     format (a,3i2,1x,i2,f6.3,3x,i3,f6.3,3x,f7.1,1x,i1)
c      seconds = im * 60 + is
c      if (startTIME .eq. 0) then
c          startTIME = seconds
c          ih0 = ih
c      endif
c      if (ih .gt. ih0) incH = 3600 * (ih-ih0)
c      itime = incH + (seconds - startTIME) + 1
c      if (ideg .ne. 34) incLAT = 60 * (ideg - 34)
c      dellAT = incLAT + (dmin - 6.733)
c      if (ideglong .ne. 119) incLONG = 60*(ideglong - 119)
c      dellONG = incLONG + (dminlong - 6.9564)
c      delALT = alt - 12.71
c      write (2,60) itime,dellAT,dellONG,delALT,isrc
60     format (1x,i6,5x,f6.3,5x,f6.3,5x,f6.1,5x,i1)
c      endif
c      goto 100
500   close (1)
c      close (2)
c      end
```

unpkvel.for:

```
c      Steve Nagengast
c      filename:  unpackvel.for
c
c      unpack mx4200 velocity data
c
c      character*11 Rtype
c      character*14 fname, outfname
c      write (*,'(a\')') ' What is the input file name?  '
c      read (*,'(a')' ) fname
c      open (1, file=fname, access='sequential',
c+status='old')
c      write (*,'(a\')') ' What is the output file name?  '
c      read (*,'(a')' ) outfname
c      open (2, file=outfname, status='new')
100   continue
        read (1,'(a)',end=500) Rtype
        write(*,'(a')' ) Rtype
        if (Rtype .eq. '$PMVXG,011,') then
          backspace 1
          read (1,50) Rtype,hdng,spd
50      format (a,f5.1,1x,f5.1)
          write (2,60) hdng,spd
60      format (1x,f5.1,5x,f5.1)
          endif
          goto 100
500   close (1)
        close (2)
        end
```

APPENDIX B. COMPUTER SIMULATION SOFTWARE

auv2surf.m:

```
% Steve Nagengast           % simulates AUV2 for initial
% filename: auv2surf.m       % two-minute surface interval

% Creates output file from AUV2 model

clear

% initial surface interval (120 seconds = two minutes)

desireddepth = 0.5;
planegain = 0.008;
idt = 1;
rpm = 550;
state(:,1) = zeros(12,1);
for k=1:120/idt
    plane = planegain * (desireddepth - state(9,k));
    if (plane > 0.7) plane = 0.7; end
    if (plane < -0.7) plane = -0.7; end
    inpt = [0; 0; plane; -plane; rpm];
    oldstate = state(:,k);
    state(:,k+1) = auv2(oldstate, inpt, idt);
end

% save and plot results from end of initial surface interval

save srfstate state

plot(-state(9,:))
```

auv2dive.m:

```
% Steve Nagengast           % simulates AUV2 thru thirty dives
% filename: auv2dive.m        %

% reload last state from initial surface interval

clear
load srfstate
stait(:,1) = state(:,121);
clear state
```

```

% run through thirty dives

numbdives = 30; idt = 1; T = 120;
rpm = 550; nn = 0;
for k2=1:numbdives
    for jj=1:T/idt
        numb = (k2-1-nn)*T + jj;
        plane = 0.02 * sin(pi*jj/60);
        inpt = [0; 0; plane; -plane; rpm];
        oldstate = stait(:,numb);
        stait(:,numb+1) = auv2(oldstate, inpt, idt);
    end
    if (k2==5)
        nn = k2; statemp=stait(:,numb+1);
        save state1 stait
        clear stait
        stait(:,1) = statemp;
    end
    if (k2==10)
        nn = k2; statemp=stait(:,numb+1);
        save state2 stait
        clear stait
        stait(:,1) = statemp;
    end
    if (k2==15)
        nn = k2; statemp=stait(:,numb+1);
        save state3 stait
        clear stait
        stait(:,1) = statemp;
    end
    if (k2==20)
        nn = k2; statemp=stait(:,numb+1);
        save state4 stait
        clear stait
        stait(:,1) = statemp;
    end
    if (k2==25)
        nn = k2; statemp=stait(:,numb+1);
        save state5 stait
        clear stait
        stait(:,1) = statemp;
    end
    if (k2==30)
        nn = k2; statemp=stait(:,numb+1);
        save state6 stait
        clear stait
        stait(:,1) = statemp;
    end
end

```

plotdpth.m:

```
% Steve Nagengast % plots the depth during first five dives
% filename: plotdpth.m

clear
load state1 % from auv2dive.m
!del divedpth.met
numdives = 5;
T = 120;
idt = 1;
for k2 = 1:numdives
    for jj = 1:T/idt
        numb = jj + (k2-1) * T;
        divedist(numb) = stait(7,numb);
        divedepth(numb) = -stait(9,numb);
    end
end

% plot results

clc
divemin = 1/60:1/60:(numdives)*120/60;
n=numdives;
subplot(211), plot(divemin,divedepth);
title('DEPTH vs. TIME');
xlabel('time, minutes');
ylabel('depth, meters');
grid;
subplot(212), plot(divedist,divedepth);
title('DEPTH vs. DISTANCE');
xlabel('distance, meters');
ylabel('depth, meters');
grid
meta divedpth;
```

simlsurf.m:

```
% Steve Nagengast % simulates INS/GPS Kalman filter for
% filename: simlsurf.m % initial 2-minute surface interval

% Estimates gyro drift errors and accelerometer biases
% in body coordinates. Also estimates errors in attitude,
% position, and velocity in earth coordinates.

% initialize state, estimated state, and covariance matrices

clear
x = zeros(15,1);
xhat = zeros(15,1);
pxxhat = zeros(15,1);
```

```

% initial variances in P(0) matrix
prep = zeros(15);
prep(1,1) = 2.35e-7;                                % radians^2
prep(2,2) = 2.35e-7;                                % radians^2
prep(3,3) = 2.35e-7;                                % radians^2
prep(4,4) = 400;                                    % meters^2
prep(5,5) = 400;                                    % meters^2
prep(6,6) = 400;                                    % meters^2
prep(7,7) = 1e-8;                                   % (meters/second)^2
prep(8,8) = 1e-8;                                   % (meters/second)^2
prep(9,9) = 1e-8;                                   % (meters/second)^2
prep(10,10) = 2.35e-15;                             % (radians/second)^2
prep(11,11) = 2.35e-15;                             % (radians/second)^2
prep(12,12) = 2.35e-15;                             % (radians/second)^2
prep(13,13) = 1e-8;                                 % (meters/second^2)^2
prep(14,14) = 1e-8;                                 % (meters/second^2)^2
prep(15,15) = 1e-8;                                 % (meters/second^2)^2

H = zeros(9,15);                                     % H matrix
for j=1:9
    H(j,j) = 1;
end

% initial transformation matrix from body to earth
% coordinates
% (assuming body is aligned with local-level frame)

lat = 0.64;                                         % radians north (vicinity Monterey)
long = pi - 2.13;                                    % radians east (vicinity Monterey)

Rbe = [-sin(lat)*cos(long) -sin(long) -cos(lat)*cos(long)
       -sin(lat)*sin(long)  cos(long) -cos(lat)*sin(long)
       cos(lat)               0           sin(lat)];

% initialize noise

rand('normal');
w = zeros(6,1);
v = zeros(9,1);

% standard deviations

sigmadrift = 1.74e-6;                               % radians/second
sigmabias = 6.1e-4;                                 % meters/second
sigmaatt = 1e-6;                                   % radians
sigmaposit = 2.5;                                  % meters
sigmavel = 0.05;                                  % meters/second

```

```

% variances

vardrift = sigmadrift^2;
varbias  = sigmabias^2;
varatt   = sigmaatt^2;
varposit = sigmaposit^2;
varvel   = sigmavel^2;

% Q matrix

Q = zeros(15);
Q(1,1) = 2.35e-12;
Q(2,2) = 2.35e-12;
Q(3,3) = 2.35e-12;
Q(7,7) = 1e-6;
Q(8,8) = 1e-6;
Q(9,9) = 1e-6;
Q(10,10) = 3e-12;
Q(11,11) = 3e-12;
Q(12,12) = 3e-12;
Q(13,13) = 3.7e-7;
Q(14,14) = 3.7e-7;
Q(15,15) = 3.7e-7;

% R matrix

R = zeros(9);
R(1,1) = varatt;
R(2,2) = varatt;
R(3,3) = varatt;
R(4,4) = varposit;
R(5,5) = varposit;
R(6,6) = varposit;
R(7,7) = varvel;
R(8,8) = varvel;
R(9,9) = varvel;

% initialize submatrices for state dynamics matrix (F)

omegAE = 0.7292115e-4;                                % radians/second
skewE = [ 0      omegaE    0
          -omegaE  0      0
                      0      0  ];
Tgyro = -eye(3) / (3600*40);                         % 1 / (correlation time)
Tacc  = -eye(3) / (3600*40);                         % 1 / (correlation time)
null3 = zeros(3);
G = [null3           null3
      null3           null3
      null3           null3
      eye(3)*vardrift null3
      null3           eye(3)*varbias ];

```

```

% initial surface interval (120 seconds = two minutes)

load srfstate                                % from auv2surf.m
for k=1:120
    f = (state(:,k+1) - state(:,k));
    dist(k) = state(7,k+1);
    depth(k) = -state(9,k+1);

    % update state dynamics matrix

    sbx = state(4,k+1) - Rbe(3,1) * omegaE;
    sby = state(5,k+1) - Rbe(3,2) * omegaE;
    sbz = state(6,k+1) - Rbe(3,3) * omegaE;
    skewb = [ 0      -sbz      sby
              sbz      0      -sbx
              -sby      sbx      0 ];
    Rbe = Rbe + Rbe * skewb;
    fb = f(1:3); fe = Rbe * fb;
    skewf = [ 0      fe(3)   -fe(2)
              -fe(3)   0      fe(1)
              fe(2)   -fe(1)   0 ];
    F = [skewE null3 null3   Rbe   null3
          null3 null3 eye(3)  null3 null3
          skewf null3 2*skewE null3 Rbe
          null3 null3 null3   Tgyro null3
          null3 null3 null3   null3 Tacc ];

    % approximate phi and gamma (idt = 1)

    F2 = F^2;
    F3 = F2 * F;
    F4 = F3 * F;
    phi = eye(15) + F + F2/2 + F3/6;
    gamma=(eye(15) + F/2 + F2/6 + F3/24 + F4/120) * G;

    % make some noise and create noisy measurements

    for n=1:6
        w(n) = rand;
    end
    v(1) = rand * varatt;
    v(2) = rand * varatt;
    v(3) = rand * varatt;
    v(4) = rand * varposit;
    v(5) = rand * varposit;
    v(6) = rand * varposit;
    v(7) = rand * varvel;
    v(8) = rand * varvel;
    v(9) = rand * varvel;
    x(:,k+1) = phi * x(:,k) + gamma * w;
    z = H * x(:,k) + v;

```

```

% Kalman filter

K = prep * H' * inv(H * prep * H' + R);
xhat(:,k) = prexhat + K * (z - H * prexhat);
p = (eye(15) - K * H) * prep;
prep = phi * p * phi' + Q;
prexhat = phi * xhat(:,k);
countdown = 120 - k
end

% save and plot results from end of initial surface interval

divex = x(:,121); stait = state(:,121);
save sn1 divex stait Rbe omegaE skewE Tgyro Tacc H prep R Q
prexhat G
save vars varatt varposit varvel
plot1

```

plot1.m:

```

% Steve Nagengast      % plots results of surface interval
% filename:  plot1.m

!del meta?.met
time = 1:120; min = 1/60:1/60:120/60; clg

subplot(211), plot(min,depth); title('DEPTH vs. TIME');
xlabel('time, minutes'); ylabel('depth, meters'); grid
subplot(212), plot(dist,depth); title('DEPTH vs. DISTANCE');
xlabel('distance, meters'); ylabel('depth, meters');
grid; meta meta1; clg

subplot(221), plot(time,x(1,1:120),'-',time,xhat(1,:),'--')
title('ATTITUDE ERROR'); xlabel('time, seconds');
ylabel('attitude error, radians'); grid
subplot(222), plot(time,x(2,1:120),'-',time,xhat(2,:),'--')
title('ATTITUDE ERROR'); xlabel('time, seconds');
ylabel('attitude error, radians'); grid
subplot(223), plot(time,x(3,1:120),'-',time,xhat(3,:),'--')
title('ATTITUDE ERROR'); xlabel('time, seconds');
ylabel('attitude error, radians'); grid; meta meta2; clg

subplot(221), plot(time,x(4,1:120),'-',time,xhat(4,:),'--')
title('POSITION ERROR'); xlabel('time, seconds');
ylabel('position error, meters'); grid
subplot(222), plot(time,x(5,1:120),'-',time,xhat(5,:),'--')
title('POSITION ERROR'); xlabel('time, seconds');
ylabel('position error, meters'); grid
subplot(223), plot(time,x(6,1:120),'-',time,xhat(6,:),'--')
title('POSITION ERROR'); xlabel('time, seconds');
ylabel('position error, meters'); grid; meta meta3; clg

```

```

subplot(221), plot(time,x(7,1:120),'-',time,xhat(7,:),'--')
title('VELOCITY ERROR'); xlabel('time, seconds');
ylabel('velocity error, meters/second'); grid
subplot(222), plot(time,x(8,1:120),'-',time,xhat(8,:),'--')
title('VELOCITY ERROR'); xlabel('time, seconds');
ylabel('velocity error, meters/second'); grid
subplot(223), plot(time,x(9,1:120),'-',time,xhat(9,:),'--')
title('VELOCITY ERROR'); xlabel('time, seconds');
ylabel('velocity error, meters/second'); grid; meta meta4;
clf

subplot(221),
plot(time,x(10,1:120),'-',time,xhat(10,:),'--')
title('GYRO DRIFT'); xlabel('time, seconds');
ylabel('gyro drift, radians/second'); grid
subplot(222),
plot(time,x(11,1:120),'-',time,xhat(11,:),'--')
title('GYRO DRIFT'); xlabel('time, seconds');
ylabel('gyro drift, radians/second'); grid
subplot(223),
plot(time,x(12,1:120),'-',time,xhat(12,:),'--')
title('GYRO DRIFT'); xlabel('time, seconds');
ylabel('gyro drift, radians/second'); grid; meta met5; clf

subplot(221),
plot(time,x(13,1:120),'-',time,xhat(13,:),'--')
title('ACCELEROMETER BIAS'); xlabel('time, seconds');
ylabel('acc. bias, meters/second^2'); grid
subplot(222),
plot(time,x(14,1:120),'-',time,xhat(14,:),'--')
title('ACCELEROMETER BIAS'); xlabel('time, seconds');
ylabel('acc. bias, meters/second^2'); grid
subplot(223),
plot(time,x(15,1:120),'-',time,xhat(15,:),'--')
title('ACCELEROMETER BIAS'); xlabel('time, seconds');
ylabel('acc. bias, meters/second^2'); grid; meta met6;

```

simdives.m:

```

% Steve Nagengast % simulates INS/GPS thru thirty dives
% filename: simdives.m

% (uses average forces and rotation rates)

% reload data from initial surface interval

clear
load sn1
load vars

```

```

* run through thirty dives

numbdives = 30; idt = 1; T = 120;
null3 = zeros(3);
w = zeros(6,1);
v = zeros(9,1);
Q = Q * T;
totalf = zeros(3,1);
totalrot = zeros(3,1);
load state1           % from auv2dive.m
state = stait;
clear stait
clear state1
nn = 0;
for k2=1:numbdives
    for jj=1:T/idt
        numb = (k2-1-nn) * T + jj;
        f = (state(:,numb+1) - state(:,numb)) / idt;
        totalrot = totalrot + state(4:6,numb+1);
        totalf = totalf + f(1:3);
    end
    if (k2==5)
        nn = k2;
        clear state
        load state2
        state = stait;
        clear stait
        clear state2
        k2 = 5;
        nn = k2;
    end
    if (k2==10)
        nn = k2;
        clear state
        load state3
        state = stait;
        clear stait
        clear state3
        k2 = 10;
        nn = k2;
    end
    if (k2==15)
        nn = k2;
        clear state
        load state4
        state = stait;
        clear stait
        clear state4
        k2 = 15;
        nn = k2;
    end
    if (k2==20)

```

```

nn = k2;
clear state
load state5
state = stait;
clear stait
clear state5
k2 = 20;
nn = k2;
end
if (k2==25)
nn = k2;
clear state
load state6
state = stait;
clear stait
clear state6
k2 = 25;
nn = k2;
end

% update transformation matrix and forces

averot = totalrot / T;
aveforce = totalf / T;
sbx = averot(1) - Rbe(3,1) * omegaE;
sby = averot(2) - Rbe(3,2) * omegaE;
sbz = averot(3) - Rbe(3,3) * omegaE;
skewb = [ 0 -sbz sby
          sbz 0 -sbx
         -sby sbx 0];
Rbe = Rbe + Rbe * skewb * T;
fe = Rbe * aveforce;

% update state dynamics matrix

skewf = [      0     fe(3)   -fe(2)
           -fe(3)     0     fe(1)
           fe(2)   -fe(1)     0 ];
F = [skewE null3 null3 Rbe null3
      null3 null3 eye(3) null3 null3
      skewf null3 2*skewE null3 Rbe
      null3 null3 null3 Tgyro null3
      null3 null3 null3 null3 Tacc ];

% approximate phi and gamma (T = 120)

F2 = F^2;
F3 = F2 * F;
F4 = F3 * F;
phi = eye(15) + F*T + F2*(T^2)/2 + F3*(T^3)/6;
gamma=(eye(15) + F*T/2 + F2*(T^2)/6 + F3*(T^3)/24 +
       F4*(T^4)/120)*T*G;

```

```

% make some system noise and update state

for n=1:6
    w(n) = rand;
end
divex(:,k2+1) = phi * divex(:,k2) + gamma * w;

% make some measurement noise

v(1) = rand * varatt;
v(2) = rand * varatt;
v(3) = rand * varatt;
v(4) = rand * varposit;
v(5) = rand * varposit;
v(6) = rand * varposit;
v(7) = rand * varvel;
v(8) = rand * varvel;
v(9) = rand * varvel;

% create noisy measurements and run through Kalman filter

z = H * divex(:,k2) + v;
K = prep * H' * inv(H * prep * H' + R);
divexhat(:,k2) = prexhat + K * (z - H * prexhat);
p = (eye(15) - K * H) * prep;
prep = phi * p * phi' + Q;
prexhat = phi * divexhat(:,k2);
divecountdown = numbdives - k2
end

% save and plot results from dives

save sn2 divex divexhat numbdives
plot2

plot2.m:

% Steve Nagengast           % plots results from one hour of
dives
% filename: plot2.m

clear; clg
pack
load sn2
!del met2?.met
divemin = 2:2:(numbdives)*2;
n=numbdives;

```

```

subplot(221),
plot(divemin,divex(1,1:n),'-',divemin,divexhat(1,:),'--')
title('ATTITUDE ERROR'); xlabel('time, minutes');
ylabel('attitude error, radians'); grid
subplot(222),
plot(divemin,divex(2,1:n),'-',divemin,divexhat(2,:),'--')
title('ATTITUDE ERROR'); xlabel('time, minutes');
ylabel('attitude error, radians'); grid
subplot(223),
plot(divemin,divex(3,1:n),'-',divemin,divexhat(3,:),'--')
title('ATTITUDE ERROR'); xlabel('time, minutes');
ylabel('attitude error, radians'); grid; meta met22; clg

subplot(221),
plot(divemin,divex(4,1:n),'-',divemin,divexhat(4,:),'--')
title('POSITION ERROR'); xlabel('time, minutes');
ylabel('position error, meters'); grid
subplot(222),
plot(divemin,divex(5,1:n),'-',divemin,divexhat(5,:),'--')
title('POSITION ERROR'); xlabel('time, minutes');
ylabel('position error, meters'); grid
subplot(223),
plot(divemin,divex(6,1:n),'-',divemin,divexhat(6,:),'--')
title('POSITION ERROR'); xlabel('time, minutes');
ylabel('position error, meters'); grid; meta met23; clg

subplot(221),
plot(divemin,divex(7,1:n),'-',divemin,divexhat(7,:),'--')
title('VELOCITY ERROR'); xlabel('time, minutes');
ylabel('velocity error, meters/second'); grid
subplot(222),
plot(divemin,divex(8,1:n),'-',divemin,divexhat(8,:),'--')
title('VELOCITY ERROR'); xlabel('time, minutes');
ylabel('velocity error, meters/second'); grid
subplot(223),
plot(divemin,divex(9,1:n),'-',divemin,divexhat(9,:),'--')
title('VELOCITY ERROR'); xlabel('time, minutes');
ylabel('velocity error, meters/second'); grid; meta met24; clg

subplot(221),
plot(divemin,divex(10,1:n),'-',divemin,divexhat(10,:),'--')
title('GYRO DRIFT'); xlabel('time, minutes');
ylabel('gyro drift, radians/second'); grid
subplot(222),
plot(divemin,divex(11,1:n),'-',divemin,divexhat(11,:),'--')
title('GYRO DRIFT'); xlabel('time, minutes');
ylabel('gyro drift, radians/second'); grid
subplot(223),
plot(divemin,divex(12,1:n),'-',divemin,divexhat(12,:),'--')
title('GYRO DRIFT'); xlabel('time, minutes');
ylabel('gyro drift, radians/second'); grid; meta met25; clg

```

```

subplot(221),
plot(divemin,divex(13,1:n),'--',divemin,divexhat(13,:),'--')
title('ACCELEROMETER BIAS'); xlabel('time, minutes');
ylabel('acc. bias, meters/second^2'); grid
subplot(222),
plot(divemin,divex(14,1:n),'--',divemin,divexhat(14,:),'--')
title('ACCELEROMETER BIAS'); xlabel('time, minutes');
ylabel('acc. bias, meters/second^2'); grid
subplot(223),
plot(divemin,divex(15,1:n),'--',divemin,divexhat(15,:),'--')
title('ACCELEROMETER BIAS'); xlabel('time, minutes');
ylabel('acc. bias, meters/second^2'); grid; meta met26;

```

esterr.m:

```

% Steve Nagengast           % plots error estimate errors
after dives
% filename: esterr.m

clear; clg
load sn2
!del sn?.met

divex = divex(:,1:numdives);
ester = divex - divexhat;

subplot(221), plot(ester(1,:)); title('    ATT. ERROR EST.
ERROR');
xlabel('dive number'); ylabel('radians per second'); grid
subplot(222), plot(ester(2,:)); title('    ATT. ERROR EST.
ERROR');
xlabel('dive number'); ylabel('radians per second'); grid
subplot(223), plot(ester(3,:)); title('    ATT. ERROR EST.
ERROR');
xlabel('dive number'); ylabel('radians per second'); grid;
meta sn1; clg

subplot(221), plot(ester(4,:)); title('POSITION ERROR
ESTIMATE ERROR');
xlabel('dive number'); ylabel('meters'); grid
subplot(222), plot(ester(5,:)); title('POSITION ERROR
ESTIMATE ERROR');
xlabel('dive number'); ylabel('meters'); grid
subplot(223), plot(ester(6,:)); title('POSITION ERROR
ESTIMATE ERROR');
xlabel('dive number'); ylabel('meters'); grid; meta sn2; clg

```

```

subplot(221), plot(ester(7,:)); title('VELOCITY ERROR
ESTIMATE ERROR');
xlabel('dive number'); ylabel('meters per second'); grid
subplot(222), plot(ester(8,:)); title('VELOCITY ERROR
ESTIMATE ERROR');
xlabel('dive number'); ylabel('meters per second'); grid
subplot(223), plot(ester(9,:)); title('VELOCITY ERROR
ESTIMATE ERROR');
xlabel('dive number'); ylabel('meters per second'); grid;
meta sn3; clg

subplot(221), plot(ester(10,:)); title('GYRO DRIFT EST.
ERROR');
xlabel('dive number'); ylabel('radians per second'); grid
subplot(222), plot(ester(11,:)); title('GYRO DRIFT EST.
ERROR');
xlabel('dive number'); ylabel('radians per second'); grid
subplot(223), plot(ester(12,:)); title('GYRO DRIFT EST.
ERROR');
xlabel('dive number'); ylabel('radians per second'); grid;
meta sn4; clg

subplot(221), plot(ester(13,:)); title('ACC. BIAS EST.
ERROR');
xlabel('dive number'); ylabel('meters per second squared');
grid
subplot(222), plot(ester(14,:)); title('ACC. BIAS EST.
ERROR');
xlabel('dive number'); ylabel('meters per second squared');
grid
subplot(223), plot(ester(15,:)); title('ACC. BIAS EST.
ERROR');
xlabel('dive number'); ylabel('meters per second squared');
grid;
meta sn5;

```

LIST OF REFERENCES

- [1] Milliken, R. J., and Zoller, C. J., "Principle of operation of NAVSTAR and System Characteristics," *Global Positioning System*, Vol. I, The Institute of Navigation, Washington, D.C., pp. 3-14, 1980.
- [2] Langley, Richard B., "The Mathematics of GPS," *GPS World*, pp. 45-50, July/August 1991.
- [3] Brown, Robert Grover, and Hwang, Patrick Y. C., *Introduction to Random Signals and Applied Kalman Filtering*, 2nd Ed., John Wiley and Sons, Inc., New York, 1992.
- [4] Cox, D. B., Jr., "Integration of GPS with Inertial Navigation Systems," *Global Positioning System*, Vol. I, The Institute of Navigation, Washington, D.C., pp. 144-153, 1980.
- [5] Bowditch, Nathaniel, *American Practical Navigator*, Vol. II, Defense Mapping Agency Hydrographic/Topographic Center, pp. 124-125, 1981 Ed.
- [6] Farrell, James L., *Integrated Aircraft Navigation*, Academic Press, New York, pp. 5-37, 1976.
- [7] Minkler, G., and Minkler, J., *Aerospace Coordinate Systems and Transformations*, Magellan Book Company, Baltimore, p. 34, 1990.
- [8] Wei, M., and Schwarz, K. P., "A Strapdown Inertial Algorithm Using an Earth-Fixed Cartesian Frame", *NAVIGATION, Journal of the Institute of Navigation*, Vol. 37, No. 2, pp. 153-167, Summer 1990.
- [9] Britting, K. R., *Inertial Navigation Systems Analysis*, Wiley-Interscience, New York, pp. 9-38, 1971.
- [10] Matthews, A., and Welter, H., "Cost-Effective, High-Accuracy Inertial Navigation," *NAVIGATION, Journal of the Institute of Navigation*, Vol. 36, No. 2, pp. 157-172, Summer 1989.
- [11] Goldnen, Eric L., and Chin, Gene H., "Environmental Performance of Attitude and Heading Reference Grade Fiber Optic Rotation Sensors," *Proc. Fiber Optic and Laser Sensors VII*, Eric Udd and Ramon P. De Paula, eds. SPIE, Bellingham, WA, Vol. 1169, pp. 347-357 September 1989.

- [12] Heer, C. V., "History of the Laser Gyro," *Proc. Physics of Optical Ring Gyros*, Stephen F. Jacobs, et al., eds., SPIE, Bellingham, WA, Vol. 487, pp. 2-12, January 1984.
- [13] Anderson, Dana Z., "Optical Gyroscopes," *Scientific American*, Vol. 254, No. 4, pp. 94-99, April 1986.
- [14] Matthews, A., "Utilization of Fiber Optic Gyros in Inertial Measurement Units," *NAVIGATION, Journal of the Institute of Navigation*, Vol. 37, No. 1, pp. 17-38, Spring 1990.
- [15] Moore, Emery L., "Fiber Optic Rate Sensors for High-G Environment Applications," *Proc. Fiber Optic and Laser Sensors VII*, Eric Udd and Ramon P. De Paula, eds., pp. 358-360, SPIE, Bellingham, WA, Vol. 1169, September 1989.
- [16] Aronowitz, Frederick, "The Laser Gyro," *Laser Applications*, Monte Ross, ed., pp. 133-200, Academic Press, New York, 1971.
- [17] Weber, Donald J., "A Three-Axis Monolithic Ring Laser Gyro," *NAVIGATION, Journal of the Institute of Navigation*, Vol. 35, No. 1, pp. 15-22, Spring 1988.
- [18] Wong, Richard V. C., *Development of a RLG Strapdown Inertial Survey System*, Ph. D. Dissertation, University of Calgary, Calgary, AB, Canada, 1988.
- [19] Miller, Christopher A., *An Application of Extended Kalman Filtering to a Model-Based, Short-Range Navigator for an AUV*, Master's thesis, Naval Postgraduate School, Monterey, CA, December 1991.
- [20] Ward, John R., and Strum, Robert D., *State Variable Analysis*, Programmed text, Naval Postgraduate School, Monterey, CA, p. 144, 1975.

BIBLIOGRAPHY

Aronowitz, Fred, "Ring Laser Gyro Technology: Headed in the Right Direction," *Photonics Spectra*, pp. 135-140, September 1988.

Beck, G. E., and others, *Navigation Systems: A Survey of Modern Electronic Aids*, Van Nostrand Reinhold Company, London, 1971.

Broxmeyer, Charles, *Inertial Navigation Systems*, McGraw-Hill Book Company, New York, 1964.

Gelb, Arthur, and others, *Applied Optimal Estimation*, The M.I.T. Press, Cambridge, MA, 1974.

Pitman, George R., Jr., and others, *Inertial Guidance*, John Wiley and Sons, Inc., New York, 1962.

Putnam, Rex G., Jr., *A Conceptual Design of an Inertial Navigation System for an Autonomous Submersible Testbed Vehicle*, Master's thesis, Naval Postgraduate School, Monterey, CA, September 1987.

Rickenbach, Mark Douglas, *Correction of Inertial Navigation System Drift Errors for an Autonomous Land Vehicle Using Optical Radar Terrain Data*, Master's thesis, Naval Postgraduate School, Monterey, CA, September 1987.

Walker, Lindsey Franklin, Jr., *Analysis and Simulation of an Autonomous Underwater Vehicle Control System*, Master's thesis, Naval Postgraduate School, Monterey, CA, December 1983.

Wells, David, and others, *Guide to GPS Positioning*, University of New Brunswick, Fredericton, N.B., Canada, 1987.

INITIAL DISTRIBUTION LIST

- | | | |
|-----|--|---|
| 1. | Defense Technical Information Center
Cameron Station
Alexandria, VA 22304-6145 | 2 |
| 2. | Library, Code 52
Naval Postgraduate School
Monterey, CA 93943-5002 | 2 |
| 3. | Chairman, Code EC
Department of Electrical and Computer Engineering
Naval Postgraduate School
Monterey, CA 93943-5000 | 1 |
| 4. | Professor R. Cristi, Code EC/Cx
Department of Electrical and Computer Engineering
Naval Postgraduate School
Monterey, CA 93943-5000 | 1 |
| 5. | Professor H. Titus, Code EC/Ts
Department of Electrical and Computer Engineering
Naval Postgraduate School
Monterey, CA 93943-5000 | 1 |
| 6. | LT Chris A. Miller
5516 Estrellita Del Norte, NE
Albuquerque, NM 87111 | 1 |
| 7. | Mr. Steve Nagengast
1384 Owens Ave.
Ventura, CA 93004 | 1 |
| 8. | Mr. Paul Heckman
Naval Ocean Systems Center
San Diego, CA 92152 | 1 |
| 9. | Mr. Kirk Dye
Naval Coastal Systems Center
Panama City, FL 32407-5000 | 1 |
| 10. | Technical Library
Naval Surface Warfare Center
Silver Spring, MD 20901 | 1 |
| 11. | RADM Evans, Code SEA-92
Naval Sea Systems Command
Washington, D.C. 20362 | 1 |

12. Technical Library, Code 9450
Naval Air Warfare Center, Weapons Division
Pt. Mugu, CA, 93042-5000

1